Affine Transform

By,

Prof (Dr.) P. P. Ghadekar

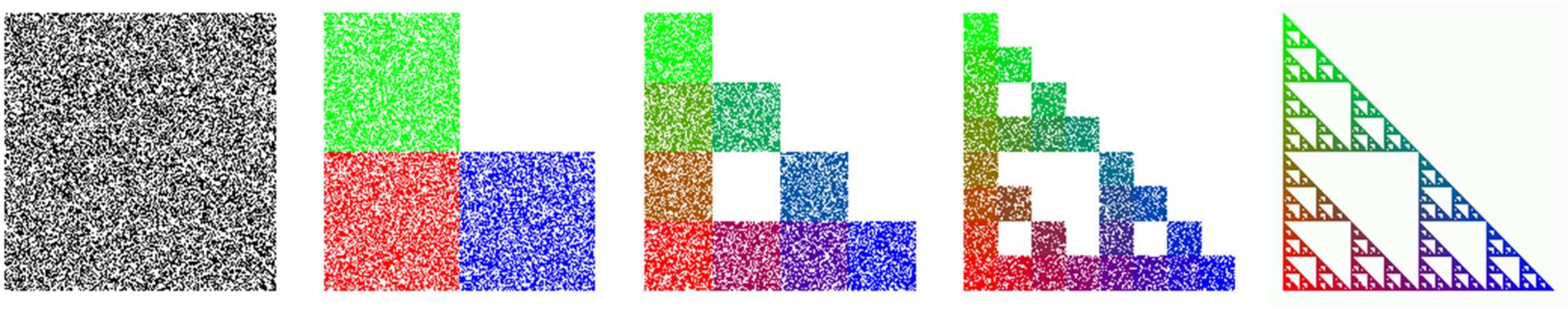
What is a Transformation?

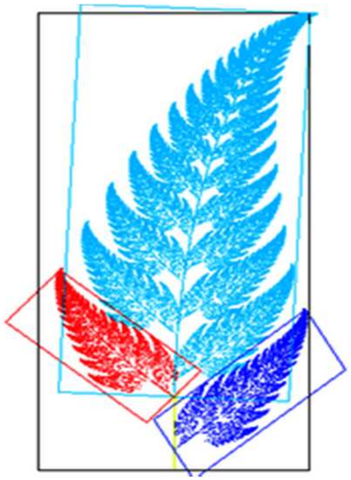
- Maps points (x, y) in one coordinate system to points (x', y') in another coordinate system

$$x' = ax + by + c$$

$$y' = dx + ey + f$$

- For example, IFS:

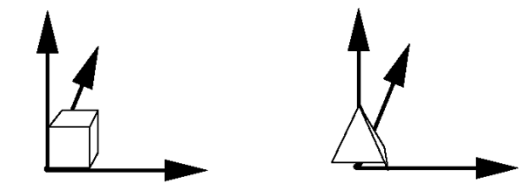


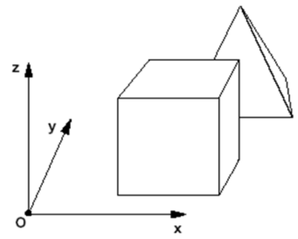


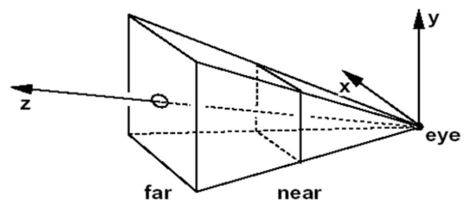
An image of a **fern-like fractal** (Barnsley's fern) that exhibits affine self-similarity.
Each of the leaves of the fern is related to each other leaf by an affine transformation.

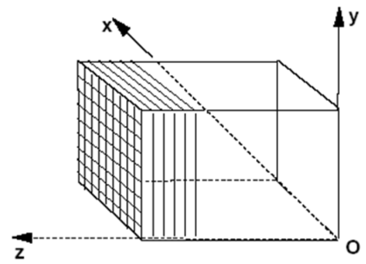
Common Coordinate Systems

- ❖ Object space
  - – local to each object
- ❖ World space
  - – common to all objects
- ❖ Eye space / Camera space
  - – derived from view frustum
- ❖ Screen space
  - – indexed according
    to hardware attributes

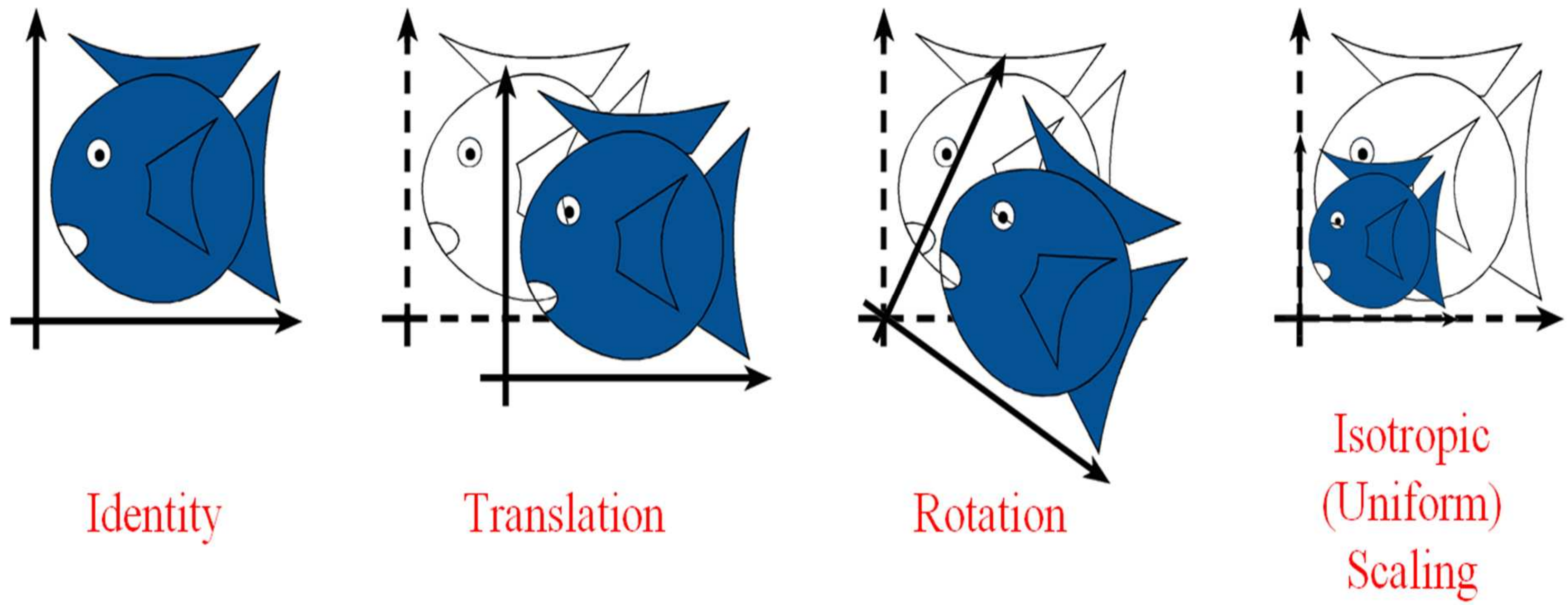








Simple Transformations



Isotropic Scaling having a physical property which has the same value when measured in different directions.

Transformations are used:

- Position objects in a scene (modeling)
- Change the shape of objects
- Create multiple copies of objects
- Projection for virtual cameras
- Animations

Affine Transform

- **Affine Transformation Describe –**
- How points, lines and planes are allowed to be transformed.
- Affine transformation is a linear mapping method that preserves points, straight lines, and planes.
- Sets of parallel lines remain parallel after an affine transformation.
- The affine transformation technique is typically used to correct for geometric distortions or deformations that occur with non-ideal camera angles.
- For example, satellite imagery uses affine transformations to correct for wide angle lens distortion, panorama stitching, and image registration.

Affine Transform

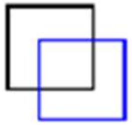
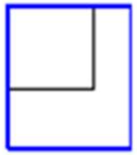
- The transformation that maps the pixel at the coordinates(x,y) to a new coordinate position is given as a pair of transformation equations.
- In this transform, straight lines are preserved and parallel lines remain unchanged.
- It is described mathematically as

$$x' = T_x(x,y)$$
$$y' = T_y(x,y)$$

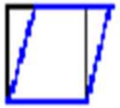
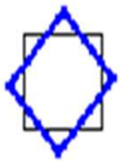
T_x and T_y are expressed as **Polynomials**. The linear equation gives an Affine Transform.

$$x' = a_0 x + a_1 y + a_2$$
$$y' = b_0 + b_1 y + b_2$$

Affine Transform

Affine Transform	Example	Transformation Matrix	
Translation		$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	t_x specifies the displacement along the x axis t_y specifies the displacement along the y axis.
Scale		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x specifies the scale factor along the x axis s_y specifies the scale factor along the y axis.

Affine Transform

Shear		$$\begin{bmatrix} 1 & sh_x & 0 \\ 1 & sh_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$	sh_x specifies the shear factor along the x axis sh_y specifies the shear factor along the y axis.
Rotation		$$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$	q specifies the angle of rotation.

Affine Transform

It is expressed in the Matrix form as

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The affine transform is a compact way of representing all transformations. The given equation represents all transformations.

Translation is the situation where $a_0 = 1$, $a_1 = 0$, and $a_2 = t_{x,}$ $b_0 = 0$, $b_1 = 1$ and $b_2 = t_{y.}$

Scaling Transformation-$a_0 = Sx$, $b_1 = Sy$ and $a_1 = 0$, $a_2 = 0$, $b_0 = 0$, $b_2 = 0$;

Rotation is a situation where $a_0 = \text{Cos}\Theta$, $a_1 = -\sin\Theta$, $b_0 = \sin\Theta$,

$b_1 = \text{Cos}\Theta$, $a_2 = 0$, $b_2 = 0$;

Horizontal shear is performed when $a_0 = 1$, $b_0 = 1$, $a_1 = Sh_x$, $a_2 = 0$, $b_1 = Sh_y$ and $b_2 = 0$.

Affine Transform Example

Affine Transformation Example :

1 Consider an image point [2,2]. Perform the following operations and show the results of these transforms.

(a) Translate the image right by 5 units.

(b) Perform a scaling operation in both x-axis and y-axis by 4 units.

(c) Rotate the image in x-axis by 45°.

(d) Perform horizontal skewing by 45°.

(e) Perform mirroring about x-axis.

Soln:- Translation of the image right by 5 units means that

$$t_x = 5, \quad t_y = 0.$$

So the translation matrix is given as

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore,

$$x' = T \times x$$

$$= \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}^T$$

$$= \begin{bmatrix} 7 & 2 & 1 \end{bmatrix}^T$$

Affine Transform Example

(b) Scaling by 4 units in bothe the directions means that

$$S = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times [2, 2, 1]$$

$$= [8\ 8\ 1]^T$$

Affine Transform Example

(c) Rotating the image in x-axis by 45°

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}^T$$

$$= \begin{bmatrix} 0 & 0.2828 & 1 \end{bmatrix}^T$$

The fraction 0.2828 is rounded to 3

$$\text{Result} = \begin{bmatrix} 0 & 3 & 1 \end{bmatrix}^T$$

Affine Transform Example

(d) Performing horizontal skewing by 45°

$$skew = \begin{bmatrix} 1 & 0 & 0 \\ \tan\theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \tan 45° & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}^T$$

$$= \begin{bmatrix} 2, & 4 & 1 \end{bmatrix}^T$$

Affine Transform Example

(e) Performing mirroring about x-axis

$$Mx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Mirroring} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 1 \end{bmatrix}$$

$$\text{Result} = \begin{bmatrix} 2 & -2 & 1 \end{bmatrix}^T$$

Thanks

Different Image File Formats

By

Prof (Dr.)P P Ghadekar

OUTLINE

- **On the completion student will be able to understand**
- Image File format Basics
- Bit Map Image(BMP)/Device Independent Bitmap File Format
- Tagged Image File Format (TIFF)
- Portable Networks Graphics (PNG) Image File Format
- Joint Photographic Image File Format (JPEG)
- Photoshop Default File Format (PSD)
- Scalable Vector Graphics File Format (SVG)

Image File Format Basics

- A digital information is often encoded in the form of binary files for the purpose of storage and transmission.
- A file format is a method used to store digital data and different file formats exists for storing images.
- Different file formats may compress the image data by differing amounts.
- Whatever be the format, the image file consists of two parts

(I) File Header

(II) Image Data

Image File Format

Format/Version
Identification
Image Width and Height
Image type
Image Data Format
Compression Type

Colour map(if any)
Pixel values

- **Header Information:-**The header part contains some of the vital information like format or version identification, width and height of the image, type of the image and image data format.

The header also specifies the type of compression mechanism.

The length of the file header is often fixed.

Bitmap File Format (BMP)

- The BMP file format is capable of storing 2D digital images of arbitrary width, height, and resolution, both monochrome and color, in various color depths, and optionally with data compression and color profiles.
- The Microsoft Windows Bitmap (BMP) file format is a basic file format for digital images in the Microsoft Windows world. BMP files have:
  - A file header
  - A bitmap header
  - A color table
  - Image data
- The file header occupies the first 14 bytes of all BMP files.

Bitmap File Format (BMP)

BMP File Header
Bitmap Information
Color Table
Pixel Data

Table - BMP file format

Extraction of BMP header

- The BMP file format .bmp is the standard file format from Windows 3.0 onwards.
- This format was developed by Microsoft for storing bitmap images in a device independent manner.
- The Windows BMP file consists of four parts:
  - File Header
  - Image Information Header
  - Color Table
  - Pixel Data

BMP file header-14 Bytes

Field Name	Size in Bytes	Description
bfType	2	The characters "BM"
bfSize	4	The size of the file in bytes
bfReserved1	2	Unused - must be zero (used for-Future Expansion)
bfReserved2	2	Unused - must be zero (used for-Future Expansion)
bfOffBits	4	Offset to start of Pixel Data

The BMP Image Information Header-40 Bytes

- A BMP information header specifies the **dimensions, compression type and color format for the bitmap.**
- The Windows BMP format is of 40 bytes long.
- The first four bytes of each format is the length of the header in bytes.

The BMP Image Information Header

Field Name	Size in Bytes	Description
biSize	4	Header Size - Must be at least 40
biWidth	4	Image width in pixels
biHeight	4	Image height in pixels
biplanes	2	Must be 1
biBitCount	2	Bits per pixel - 1, 2, 4, 8, 16, 24, or 32
biCompression	4	Compression type (0 = uncompressed)
biSizeImage	4	Image Size - may be zero for uncompressed images
biXPelsPerMeter	4	Preferred resolution in pixels per meter
biYPelsPerMeter	4	Preferred resolution in pixels per meter
biClrUsed	4	Number of Color Map entries that are actually used
biClrImportant	4	Number of significant colors. If zero then all colors are important.

The BMP Color Table- 4Bytes

- A color table is defined as an array of RGB Quad structures, contains as many elements as there are colors in the bitmap.
- There is no color table for 24 bit BMP images as each pixel is represented by 24-bit Red-Green-Blue (RGB) values in the actual bitmap data area.
- For an 8 bit bitmap image the color table consists of 256 entries with each entry consisting of four bytes of data.
- The first three bytes are the blue, green and the red colors respectively.
- The fourth byte is unused and must be equal to zero.

The BMP Color Table

Field Name	Size(bytes)	Description
RgbBlue	1	specifies the blue part of the color
rgbGreen	1	specifies the green part of the color.
rgbRed	1	specifies the red part of the color.
rgbReserved	1	must always be set to zero.

The Pixel Data

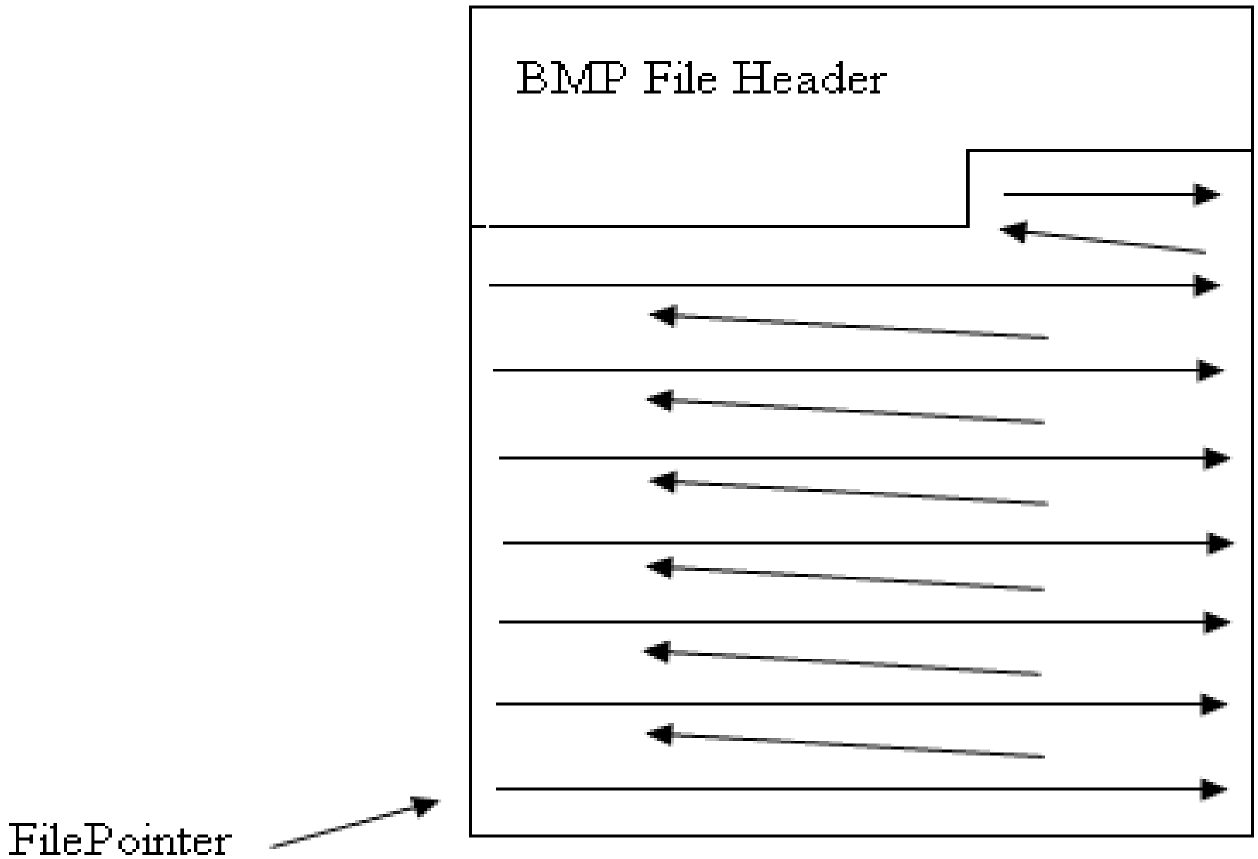
- In the 8-bit format each pixel is represented by a single byte of data, that byte is an index into the color table.
- In 24-bit image, each pixel is represented by three consecutive bytes of data that specifies the blue, green, and red component values respectively.
- In 24-bit BMP file the color table is absent.
- The pixel data is actual image data represented by consecutive rows, or scan lines.
- The number of bytes in one row must always be adjusted to fit into the border of multiple of four.

The Pixel Data

- You simply append zero bytes until the number of bytes in a row reaches of multiple of four.
- Pixels are stored "upside-down" with respect to normal order, starting in the lower left corner, going from left to right, and then row by row from the bottom to the top of the image

The Pixel Data

- The following figure illustrates the pixel data stored in a BMP image. Image is stored from bottom left corner as indicated by the file pointer and then moving upwards row by row.



Algorithms for extraction of BMP header

❖ **Using inbuilt MATLAB function.**

1. Create a new MATLAB file.
2. Use the *imfinfo* function to copy the BMP header into a variable.
3. Display the BMP header.

Algorithms for extraction of BMP header

- **Using C**

1. Define a structure for bitmap file header.
2. Define a structure for bitmap information header.
3. Open the 8 bit BMP file.
4. Read the file header of BMP file one byte at a time and store its contents in structure for file header.
5. Read image information header one byte at a time and store its contents in structure for info header.
6. Display the contents of file header and image info header on DOS screen.
7. Display image by using put pixel function using Graphics.
8. Close all file pointers and exit.

Tagged Image File Format (TIFF)

- It stands for Tagged Image File Format and was developed by the Aldus Corporation in the 1980s.
- It was later supported by Microsoft. TIFF files are often used with scanned images.
- Since a TIFF file does not compress an image file, images are often large but the quality is preserved.
- It can handle multiple images and data in a single file through the inclusion of 'tags' in the file header.
- Tags can indicate the basic geometry of the image, such as its size, or define how the image data is arranged and whether various image compression options are used and uses a filename extension of TIFF or TIF.

Tagged Image File Format (TIFF)

- The TIFF format is often used to exchange files between applications and computer platforms.
- Within TIFF, a lossless compression routine known as LZW is available. This reduces the size of the stored file without perceptible loss in quality.
- The goals of the TIFF specification include extensibility, portability and reversibility.

Portable Networks Graphics (PNG)

- PNG stands for Portable Networks Graphics.
- PNG is bitmapped image format that employs lossless data compression.
- PNG was created to improve and replace GIF format. The PNG file format is regarded and was made as free and open-source successor to the GIF file format.
- The PNG file format supports true (16 million colors) whereas the GIF format only allows 256 colors.
- PNG excels when the image has large areas of uniform colors.

Portable Networks Graphics (PNG)

- The lossless PNG format is best suited for editing pictures, whereas the lossy formats like JPEG are best for the final distribution of photographic-type images due to smaller file size.
- Yet many earlier browsers do not support the PNG file format. However, with the release of Internet Explorer 7, all popular modern browsers fully support PNG.
- Special features of PNG files include support for up to 48 bits of color information.

Joint Photographic Experts Group (JPEG)

- JPEG is not actually a file type. It is the most important current standard for image compression.
- JPEG standard was created by a working group of the International Organization of Standards (ISO).
- This format provides the most dramatic compression option for photographic images.
- JPEG compression is used with the JFIF file format that uses the file extension '.jpg'.
- This format is useful when the storage space is at a premium.
- JPEG pictures store a single raster image of 24-bit color.

Joint Photographic Experts Group (JPEG)

- JPEG is a platform-independent format that supports the highest levels of compression.
- However, this Compression is lossy. Progressive JPEG files support interlacing.

The important features of JPEG file format

- Lossy compression scheme.
- The images are not interlaced. In case of progressive JPEG, the images can be interlaced.
- The strength of JPEG is its ability to compress large image files.
- Due to this compression, the image data can be stored effectively and retransmitted efficiently from one place to another.
- The base version of JPEG does not support multiple layers and high dynamic range.
- Hence JPEG will not be a wise choice if one is interested in maintaining high quality pictures.

Comparison

Attribute	JPG	GIF	PNG	BMP
Compression	Lossy	Lossless	Lossless	Lossless
Bits	8 or 24 bits	24 bits	8 or 24 bits	1,4,8,16,24,32,48, or 64 bits
Colour	Not Transparent	Transparent	Better Transparent	Not Transparent

Transparent Meaning

- A transparent GIF (Graphics Interchange Format) is an image file that has one color assigned to be "transparent" so that the assigned color will be replaced by the browser's background color, whatever it may be.
- For example, that you have created a rectangular GIF image of a large red star on a white background.
- If you are only interested in having the red star appear on your Web page, and don't want to see the white background, you can transparentize the white background color so that it changes to whatever the Web page's background color is (yellow, for example).
- Then, when you view the Web page, you will only see a red star on a yellow background.

How To display a Bitmap Image in C and MATLAB

- **ALGORITHMS**
- **Using MATLAB**
  - Read the image using the ***imread*** **function.**
  - Display the image using the ***imshow*** **function.**
- **Using C-Following C Functions are Used**
- initgraph
- Graphresult
- Getpalette
- Setrgbpalette
- Putpixel
- Settextstyle
- outtextxy

Function	What it does
initgraph	Initialize the graphics system by loading a graphics driver from disk (or validating a registered driver) and then putting the system into graphics mode.
graphresult	Returns the error code for the last graphics operation that reported an error, then resets the error level to gr0k.
getpalette	Fills the palette type structure *palette with information about the current palette's size and colors.
setrgbpalette	Defines colors for IBM-8514 and VGA graphics card drivers. Only the lower byte of red, green, or blue is used, and out of each byte, only the 6 most significant bits are loaded in the palette (Therefore, we shift left 2 bits by multiplying by 4).
putpixel	Plots a pixel at a specified point.
settextstyle	Sets the text font, the direction in which text is displayed, and the size of the characters.
outtextxy	Displays a string at the specified location.

Using C-Algorithm

1. Use *initgraph* to initialize the graphics system by requesting auto detection of graphics driver and setting highest resolution for the detected driver.

2. Use *getpalette* to get information about the current palette's size and colors.

3. Use *setrgbpalette* to define colors for the graphics card. Since only the 6 most significant bits of the lower byte of red, green and blue are loaded in the palette, multiply each by 4 to shift each value left by 2 bits.

4. Get the image source file from the user and open the file in read binary mode.

5. Get the width and height of image from the BMP file header.

Using C-Algorithm

6. Set the file pointer to the pixel data and store each pixel in an array. The number of bytes in one row of the pixel are always adjusted to fit into the border of multiple of four. Thus, if the width of the image is not a multiple of four, skip the remaining bytes.

7. Display the string 'Bitmap Image' on the screen using the *settextstyle* and the *outtextxy* functions.

8. Use the *putpixel* function to display each pixel on the screen in upside-down order (to compensate for the upside-down storing of the pixel data).

9. Clear the graphics screen, close all file pointers and exit.

JPEG-2000

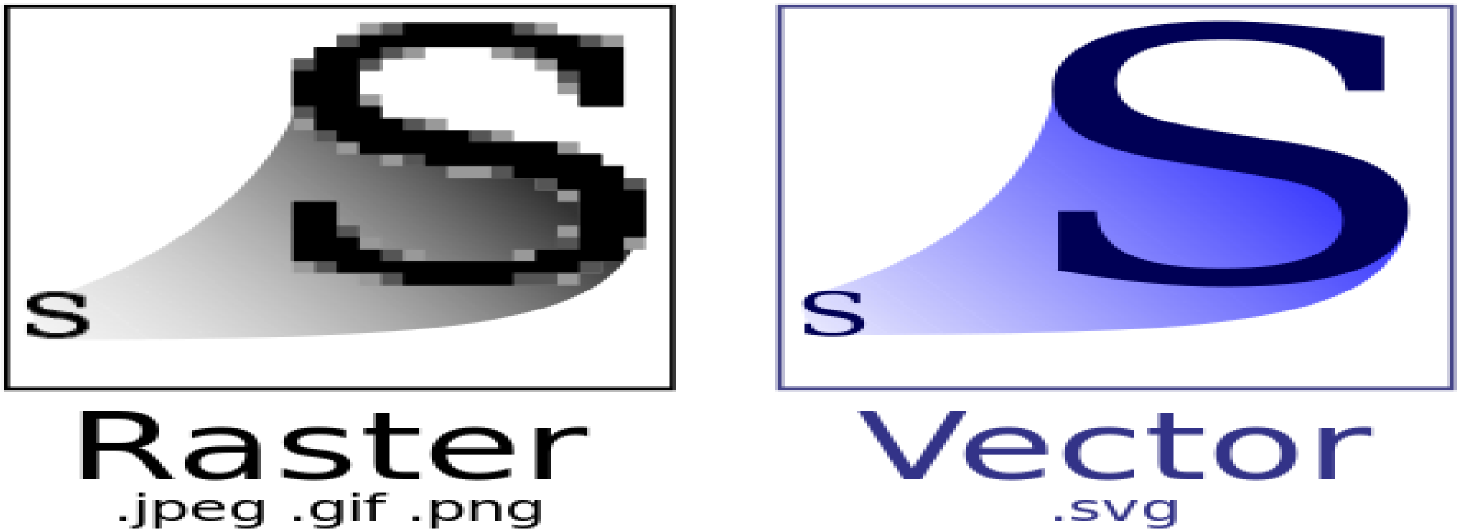
- The JPEG committee has released a new version of their file format called JPEG 2000.
- JPEG 2000 employs wavelet-based image compression algorithm whereas JPEG employs DCT based image compression.
- JPEG 2000 offers progressive image downloading.
- The progressive image downloading allows the user to download a lower resolution version of an image and to continue downloading a more detailed version if needed.

Photoshop Default File Format (PSD)

- PSD is the Photoshop's default format which is capable of supporting layers, editable text and millions of colours.
- It has the capacity to retain information related to layers, masking channels etc. , which tend to be lost when saved in another file format.
- The PSD format has no compression features, but should still be used to archive complex images with multiple layers and sophisticated selections or paths.
- Photoshop 5.5 has a lossless compression routine built into it.

Scalable Vector Graphics File Format (SVG)

- It is used for Internet use.
- This format is specifically based on XML, intended to create a smaller, scalable image file format for internet use.
- It is generally smaller than bitmapped file and scalable without any loss of resolution.
- The information for the image is stored in the text data so that it can be searched and edited.



Thanks

Hough Transform & Harris Corner Detection

By,

Prof(Dr.) Premanand P Ghadekar

Outline

- ❖ Edge Linking and Boundary Detection
- ❖ Introduction to Hough Transform
- ❖ Straight line fitting
- ❖ Hough Transform Algorithm
- ❖ Updated Hough Transform Algorithm
- ❖ Harris Corner Detection

Image Feature Extraction-Recapitulation

- Edges (edge pixels)
- Sobel, Roberts, Prewitt
- Laplacian of Gaussian
- Canny Edge detector
- Watershed Segmentation
- Region Growing
- Split & Merge Algorithm etc.

Shape Features

- Straight Lines
- Circles and Ellipses
- Arbitrary Shape

Edge Linking and Boundary Detection

❖ Edge Linking Procedures assemble edge points into meaningful boundaries.

Edge Linking

Local Processing **Global Processing**

Local Processing

- Take an Edge detected Image
- Analyze each pixel in a small neighborhood of every point (x,y) .
  - All points that are similar in nature are linked
  - This forms a Boundary of pixels that are similar in nature.

- Similarity
  - Strength of the response of gradient operator
  - Direction of the gradient

Local Processing

- Edge pixels (x,y) & (x', y') are similar if

$$|\nabla f(x,y) - \nabla f(x',y')| \leq T$$

$\nabla f(x,y)$-Gradient at x,y & $\nabla f(x',y')$-Gradient at x',y'

$$|\alpha(x,y) - \alpha(x', y')| \leq A$$

$\alpha(x,y)$- Direction at (x,y) & $\alpha(x',y')$ –Direction at (x',y')

$$(x',y') \in N_{xy}$$

T is non-negative threshold & A is Angle threshold

Global Processing-Hough Transform

- If (x’, y’) is **not with in the neighborhood** of (x,y), then (x’,y’) can not be linked to (x,y). **Local processing technique does not help.**
- **Global Processing technique –Hough Transform** can be used
- Method and Means for Recognizing Complex Patterns, Paul V. C. Hough et al (Patent)
- Hough Transform is the **Mapping from Spatial domain to Parameter space**

Hough Transform

❖ **Elegant method for Direct Object Recognition**

- The **Hough transform** is a feature extraction technique used in image analysis, computer vision, and digital image processing.
- The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by **a Voting Procedure.**
- This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called **Accumulator Space** that is explicitly constructed by the algorithm for computing the Hough transform.
- The classical Hough transform was concerned with the **identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses.**

Hough Transform

- The Hough transform takes the images created by the edge detection operators.
- Most of the time, the edge map generated by the edge detection algorithms is disconnected.
- The Hough transform can be used **to connect the disjointed edge points**.
- Unlike the edge linking algorithms, the Hough transform does not require any prerequisites to connect the edge points.
- It is used **to fit the points as plane curves.**
- The plane curves typically **are lines, circles, and parabolas.**
- The equation of a line is given below

$$y=mx+c$$

Where **m** is the slope of the line, **c** is **y** intercept of the line.

Image and Parameter Spaces

Equation of Line: $y = mx + c$

Find: (m, c)

Consider point: (x_i, y_i)

$$y_i = mx_i + c \quad or \quad c = -x_i m + y_i$$

Parameter space also called **Hough Space**

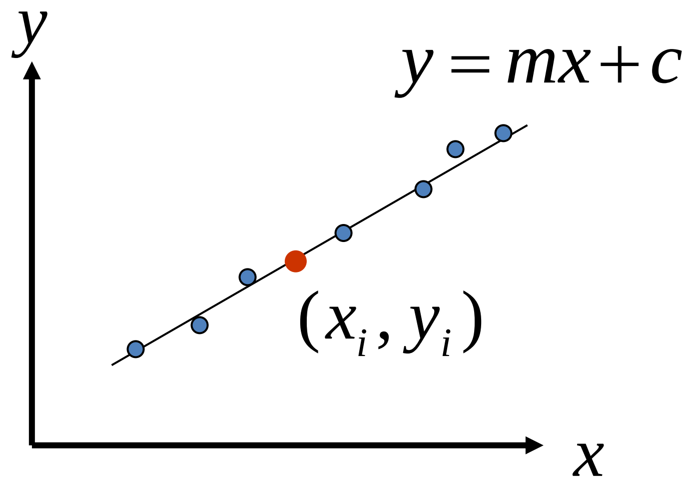
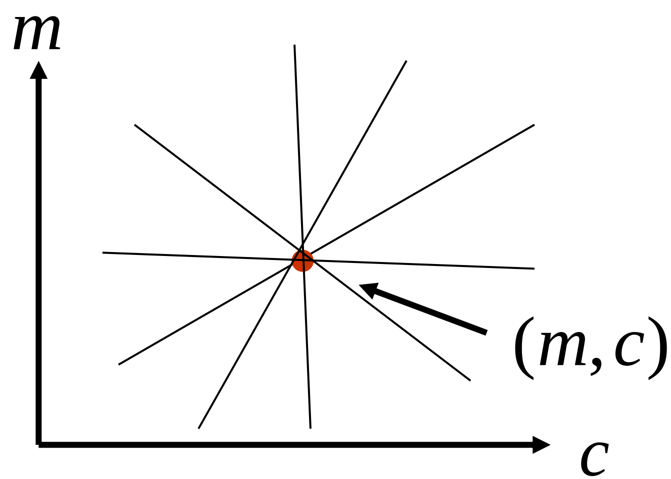


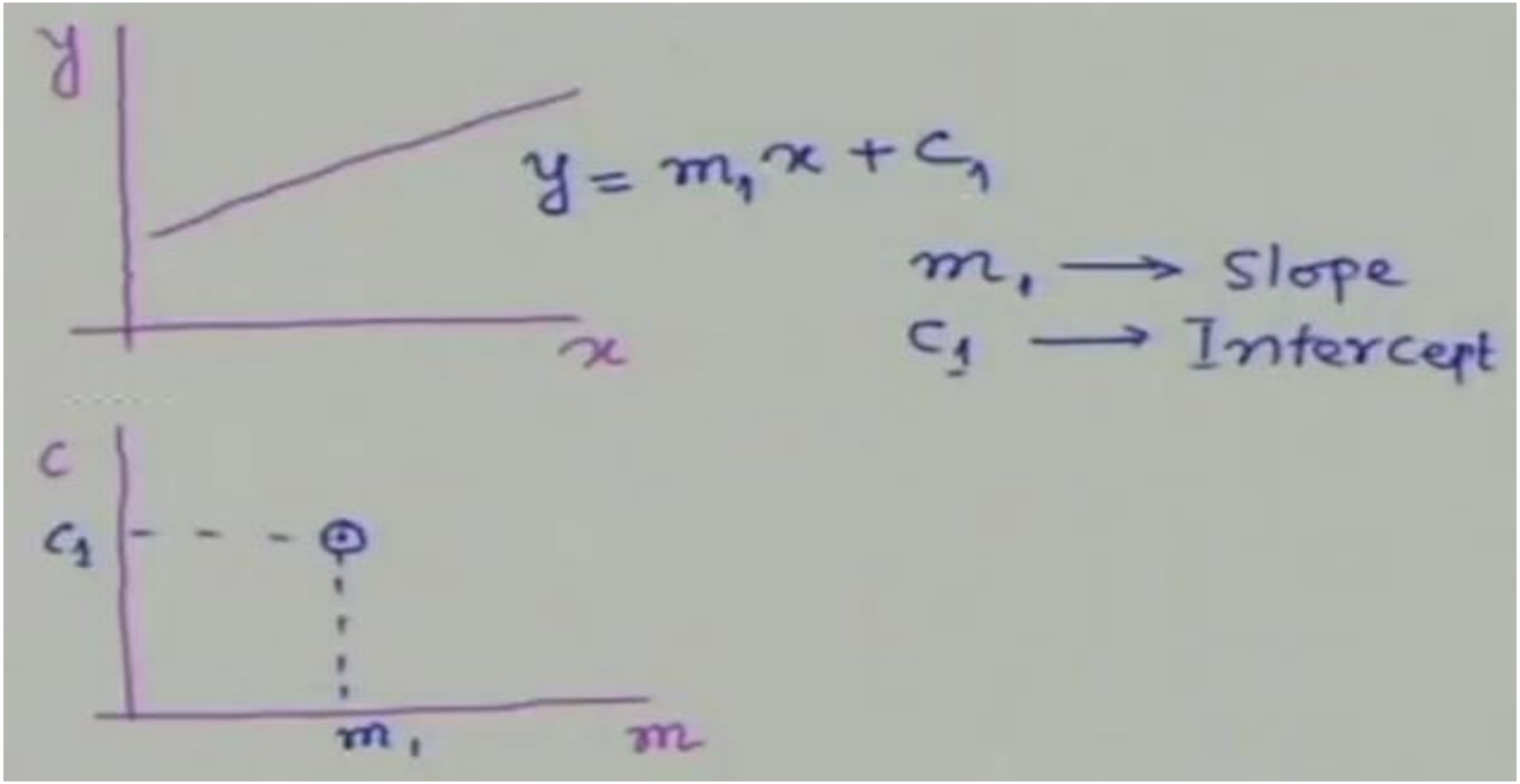
Image Space



Parameter Space

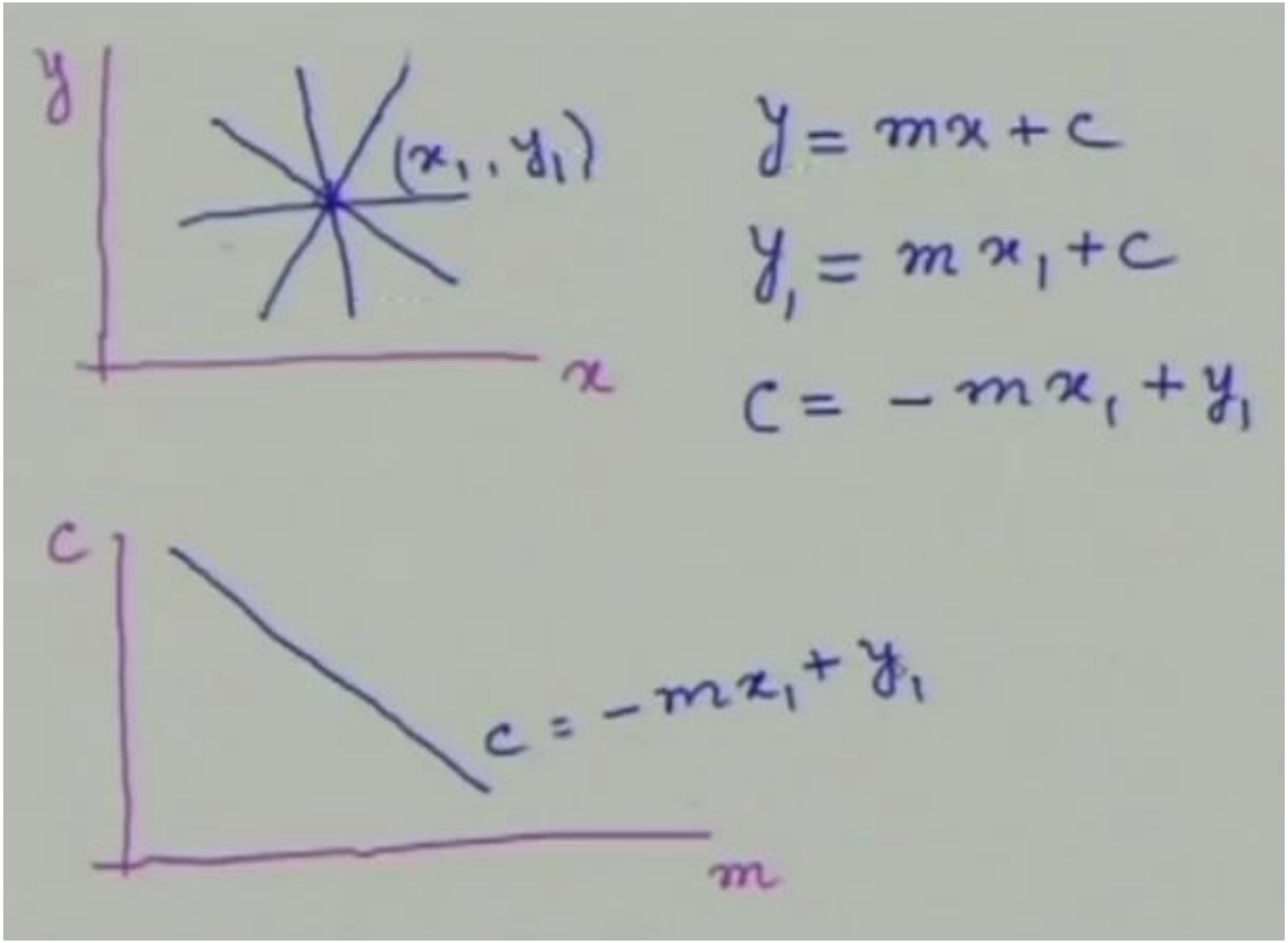
Image and Parameter Spaces-Line

An edge point in x-y plane is transformed to a c-m plane. Line eq- **y=mx+c**
Write the line equation as **c=-mx+y**



A set of **points** are said to be **collinear** if they all lie on a single line. Because there is a line between any two **points**, every pair of **points** is **collinear**.

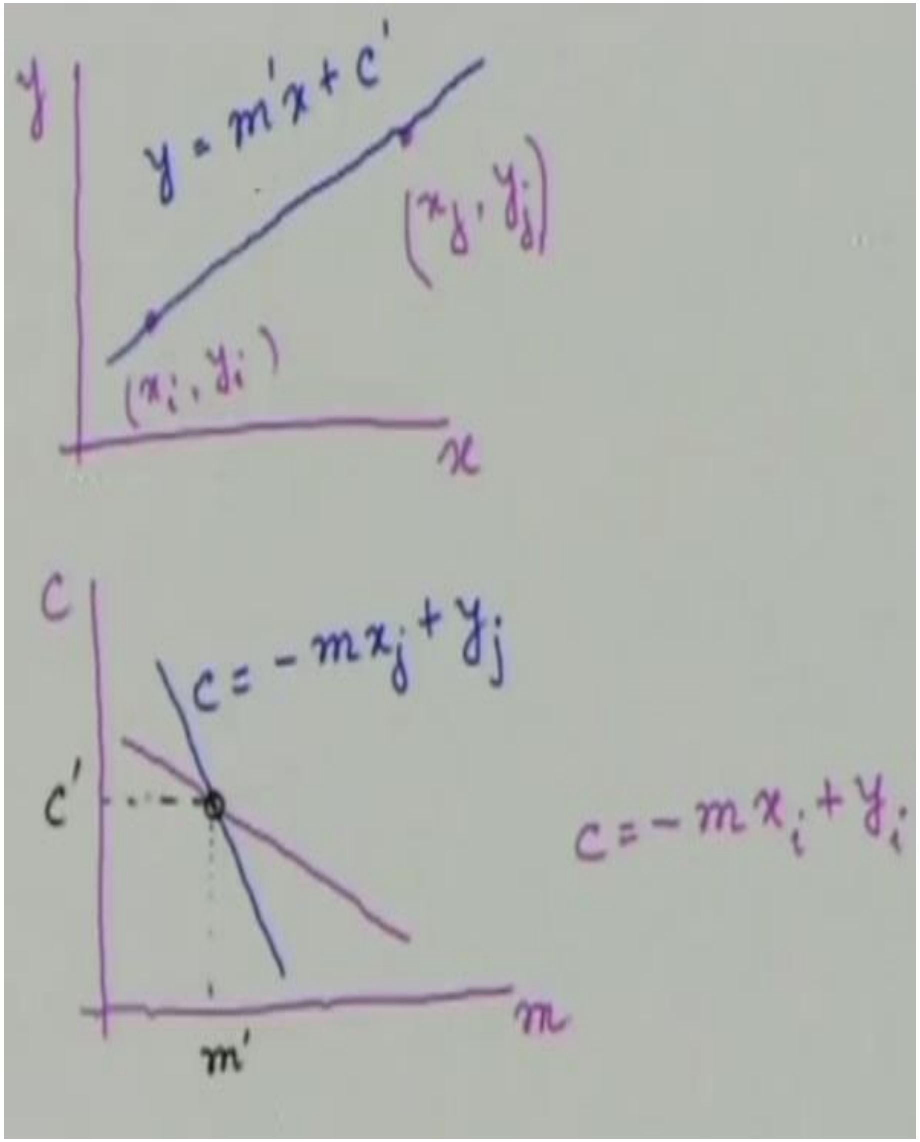
Image and Parameter Spaces-Point



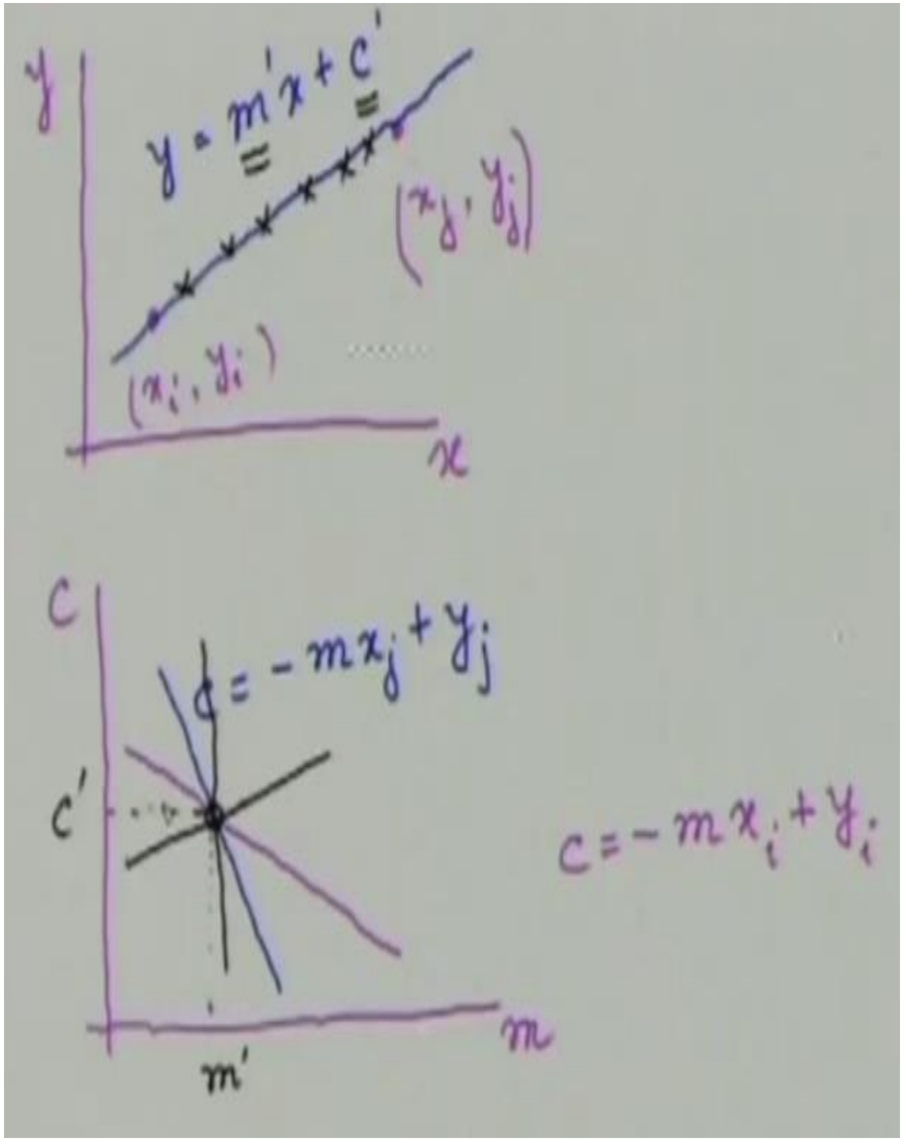
Hough Transform

- All the edge points (xi,yi), (xi+1,y_{i+1})…….(x_j,y_j) in the x-y plane need to be fitted .
- Hence the x-y plane should be transformed into a different c-m plane.
- All points are lines in the c-m plane.
- The objectives is to find the intersection point.
- A common intersection point indicates that the edge points are part of the same line.
- If A & B are points connected by a line in the spatial domain, then they will be intersecting lines in the Hough Space.

Image and Parameter Spaces-Two Points

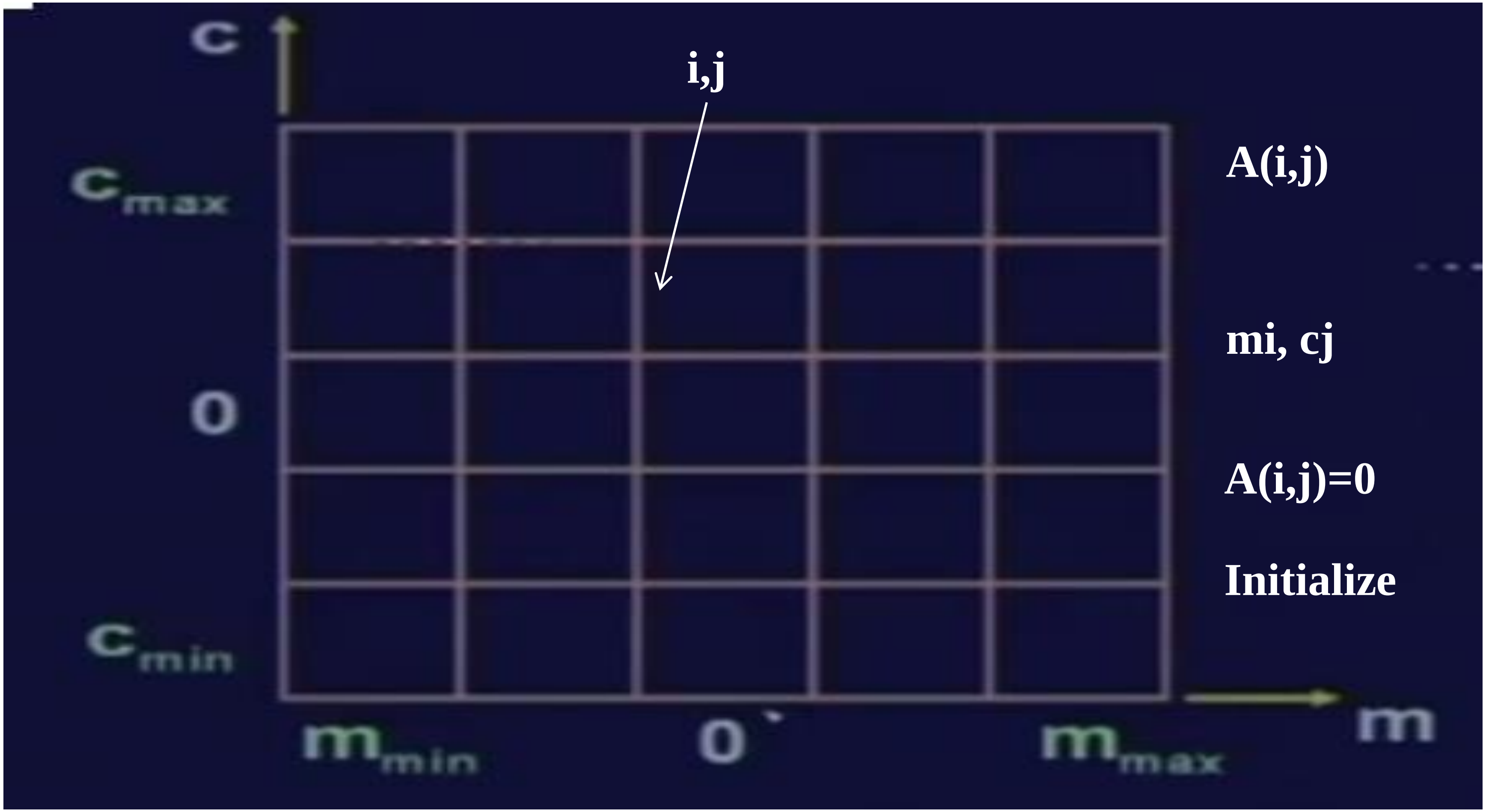


(a)



(b)

Accumulator Array-Digital –c-m plane



Where- m_{min} is minimum Slope value & m_{max} is the maximum slope value. C_{min} is the minimun intercept value of c and C_{max} is the maximum intercept value of c

Hough Transform Algorithm

❖ **The Hough Transform is stated as follow**

1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image :-

   If the pixel (x_k , y_k) is an edge pixel, then

(a) $c=(-m)x_k + y_k$

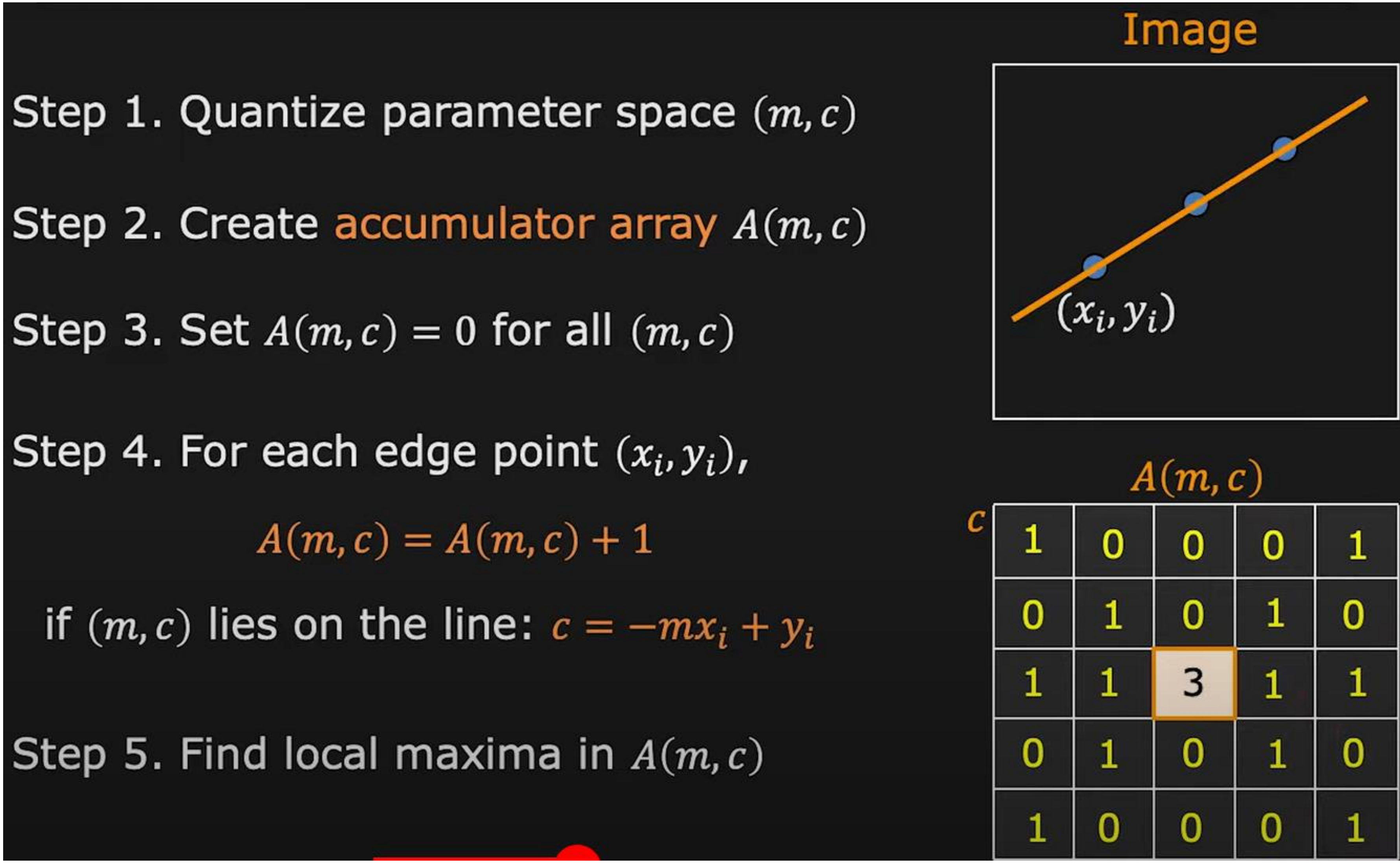
(b) $A(p, q)=A(p, q)+1$

5. Show the Hough Space.

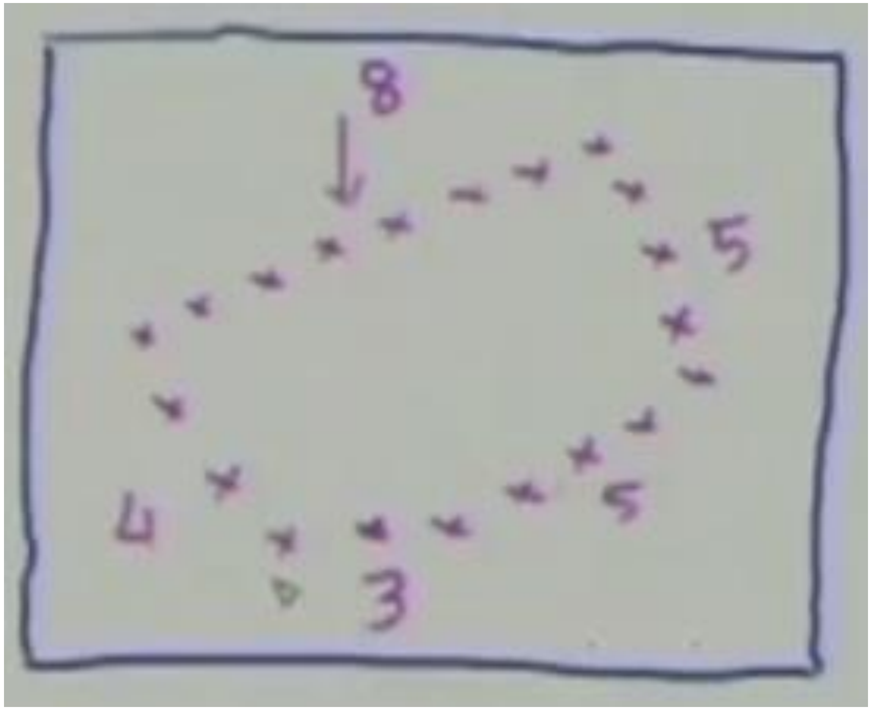
   $A(i,j)=Q$

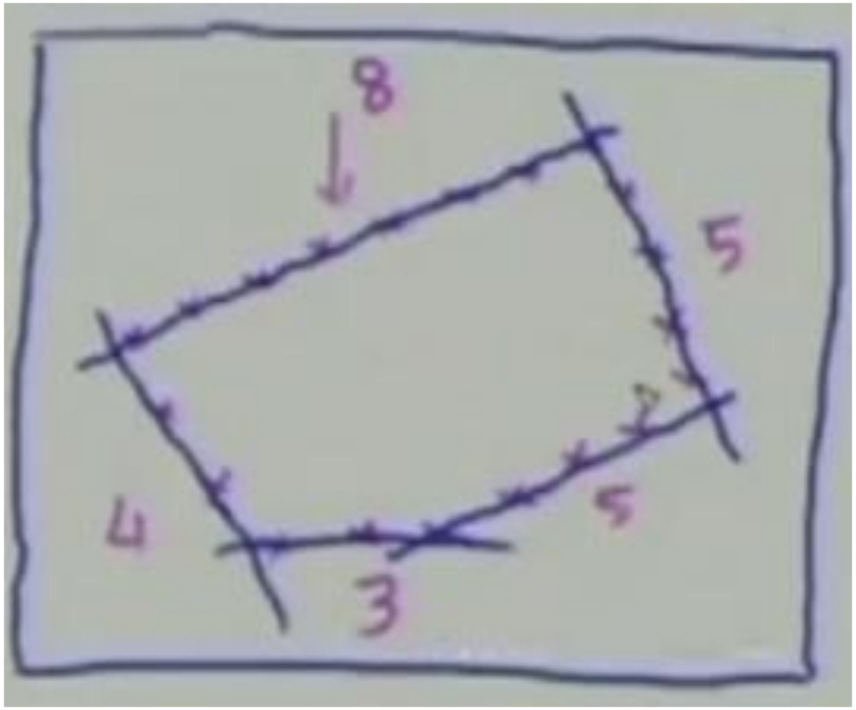
6. Find the local maxima in the parameter space
7. Draw the line using the local maxima.

Accumulator Array-Digital –c-m plane



Hough Transform-Line Detection Example





- ❖ **Problem-It does not work for vertical lines, as they have a slope of infinity. Therefore, it is better to convert this line in to polar coordinates.**

Modified Hough Transform

- The line in polar coordinates form can be represented as ρ=x cosΘ +y sin Θ, where Θ is the angle between the line and x-axis, and ρ is the diameter.
- **The Modified Hough Transform-**

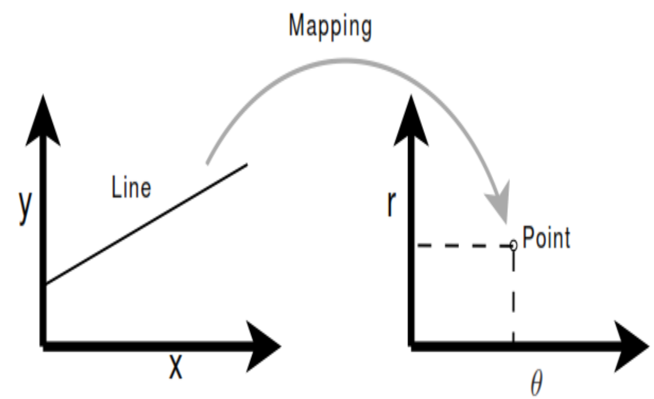
1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image:-

   If the pixel is an edge pixel, then for all Θ

   (a) Calculate ρ for the pixel, then for all Θ

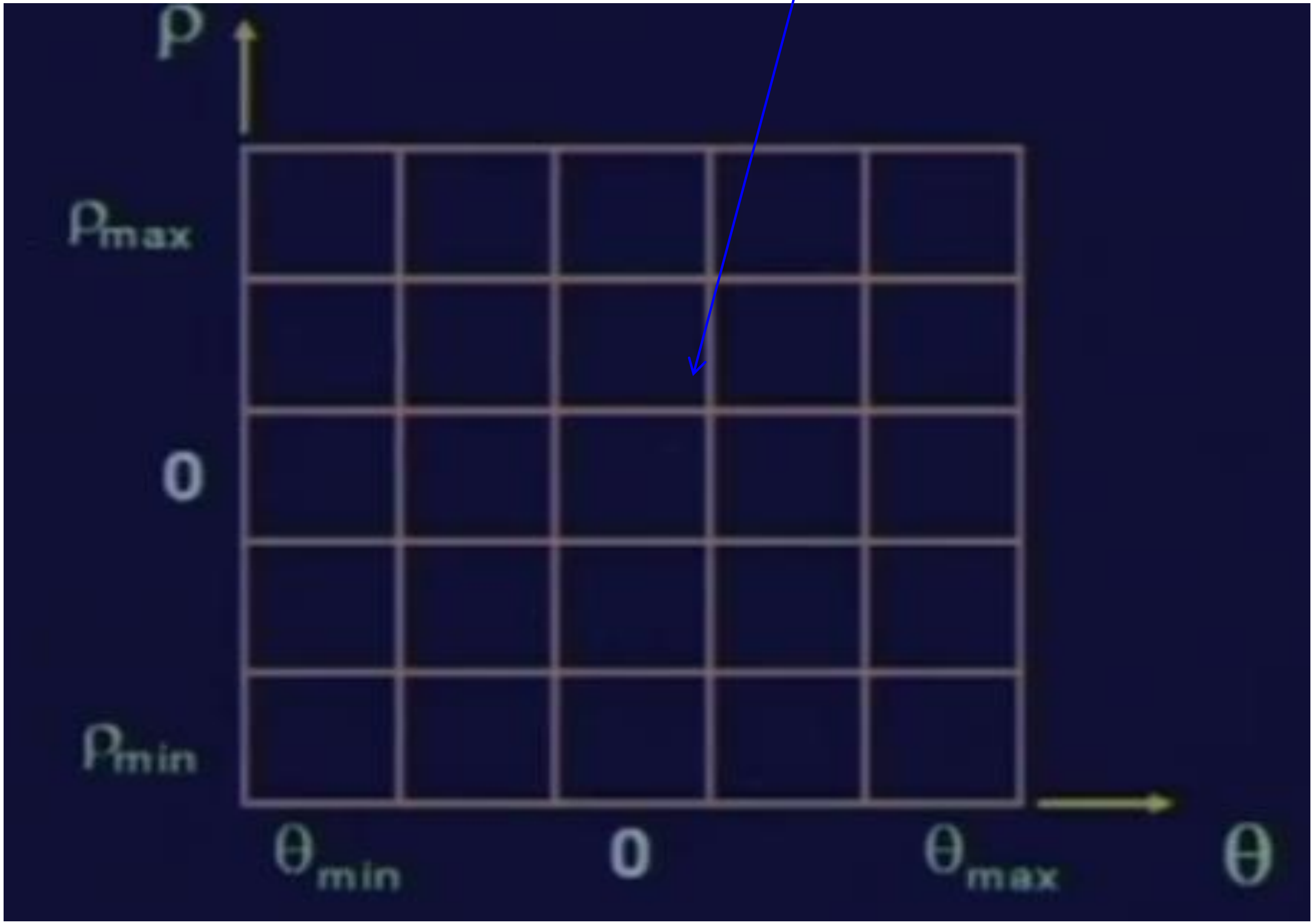
   (b) Increment the position (ρ,Θ) in the accumulator array *P*.

5. Show the Hough Space.
6. Find the local maxima in the parameter space.
7. Draw the line using the local maxima.



Modified Hough Transform

(i,j)

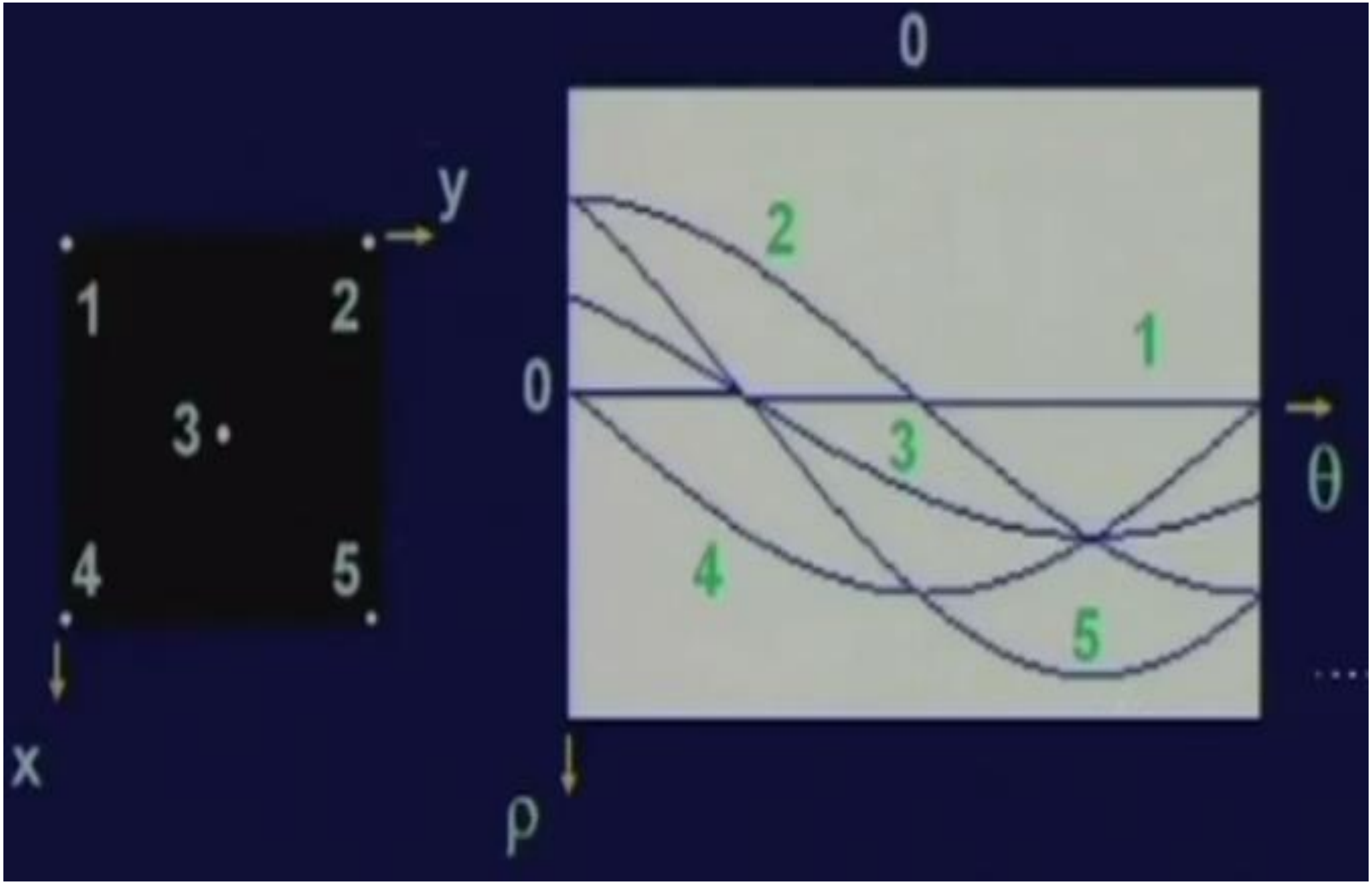


A(i,j)

θi, ρj

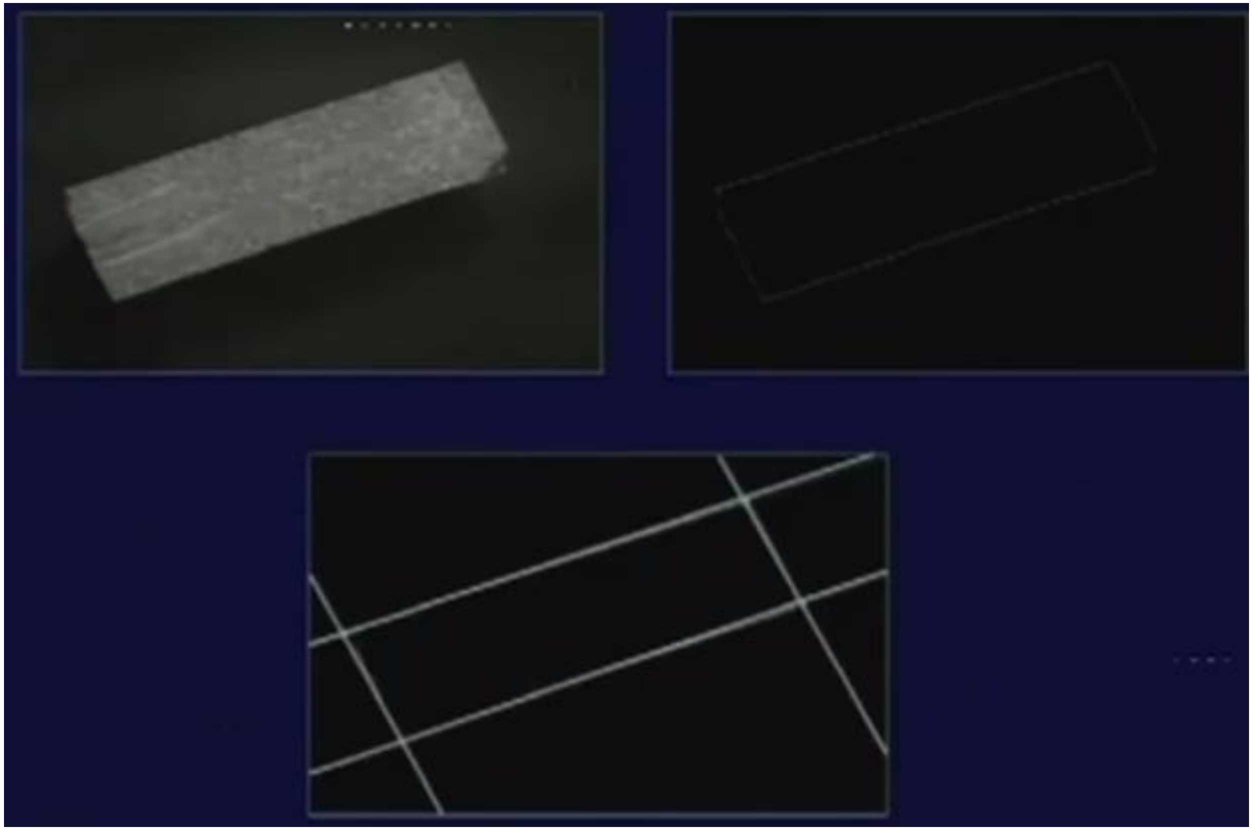
$\rho = X_k \cos\theta + Y_k \sin\theta$

Modified Hough Transform



Hough Transform

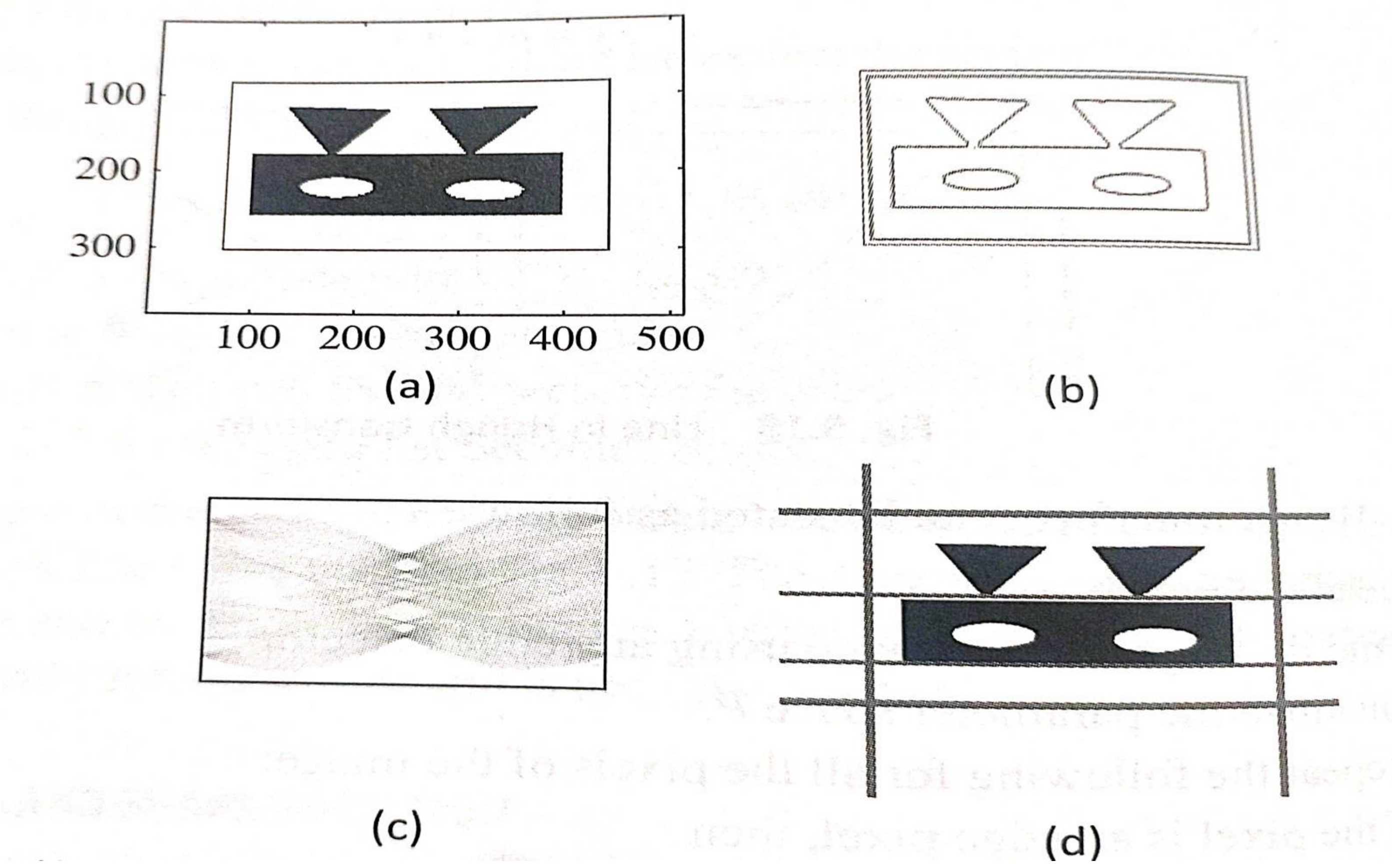
❖ **The result of the Hough Transform line detection is shown below**



(a) Input image (b) Output Edges (c) Detected Lines

Hough Transform

❖ **The result of the Hough Transform line detection is shown below**



(a) Input image (b) Output Edges (C) Accumulator Image (D) Detected Lines

Hough Transform-Circle Equation

A circle with center at (3,4) and a radius of 6:

Soln- Circle Equation-

$$(x-a)^2 + (y-b)^2 = r^2$$

Put in (a, b) and r:

$$(x-3)^2 + (y-4)^2 = 6^2$$

Hough Transform for Circle Detection

- ❖ The Hough Transform for other shapes can also be found.
- ❖ The Hough Transform for a **circle detection** can be given as follows:-

$$(x-a)^2 + (y-b)^2 - r^2 = 0$$

The parameter space is three dimensional as the unknown are a, b, and r.

The polar form Circle detection algorithm is-

1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image:-

   If the pixel is an edge pixel, then for all values of r, Calculate

(a) $a = x - r\cos\Theta$

(b) $b = y - r\sin\Theta$

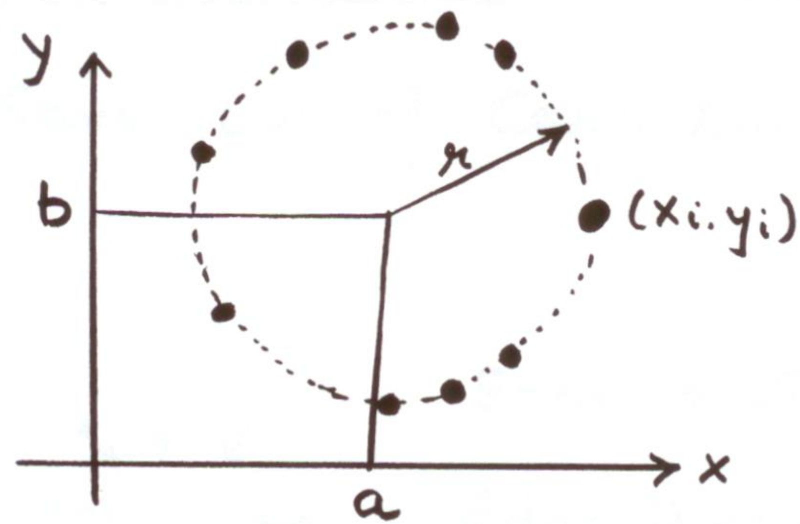
(c) $A(a, b, r) = A(a, b, r) + 1$

5. Show the Hough Space.
6. Find the local Maxima in the parameter space.
7. Draw the Circle using the local Maxima.

Finding Circles by Hough Transform

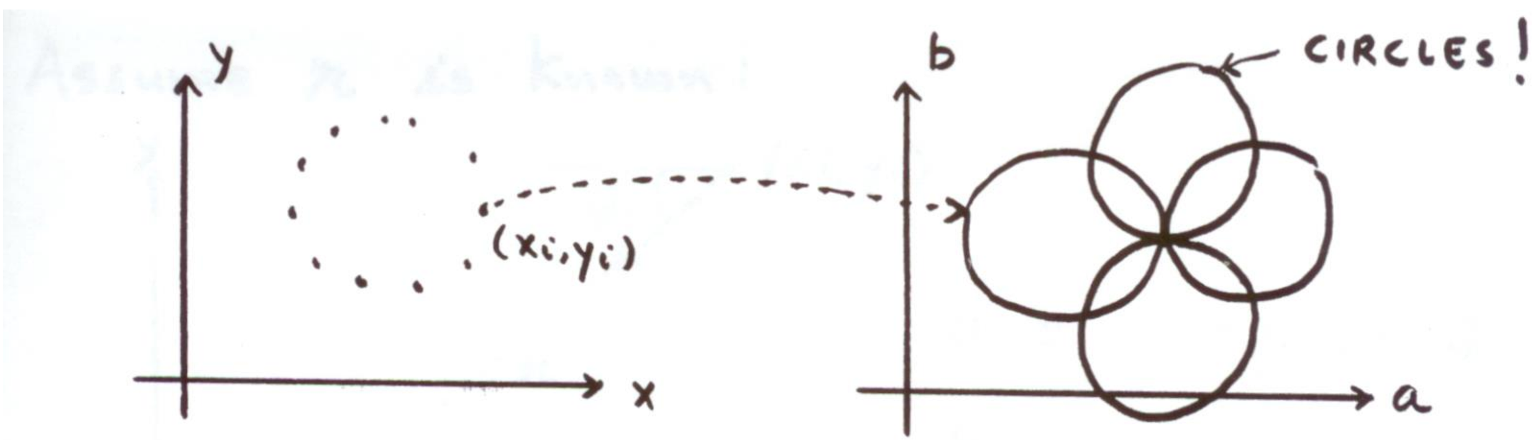
Equation of Circle:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$



If radius is known: (2D Hough Space)

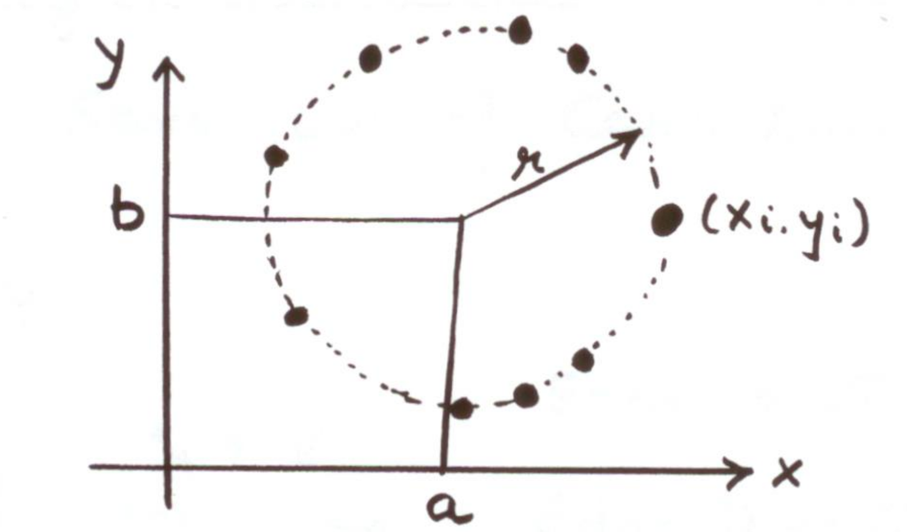
Accumulator Array $A(a, b)$



Finding Circles by Hough Transform

Equation of Circle:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$



If radius is not known: 3D Hough Space!

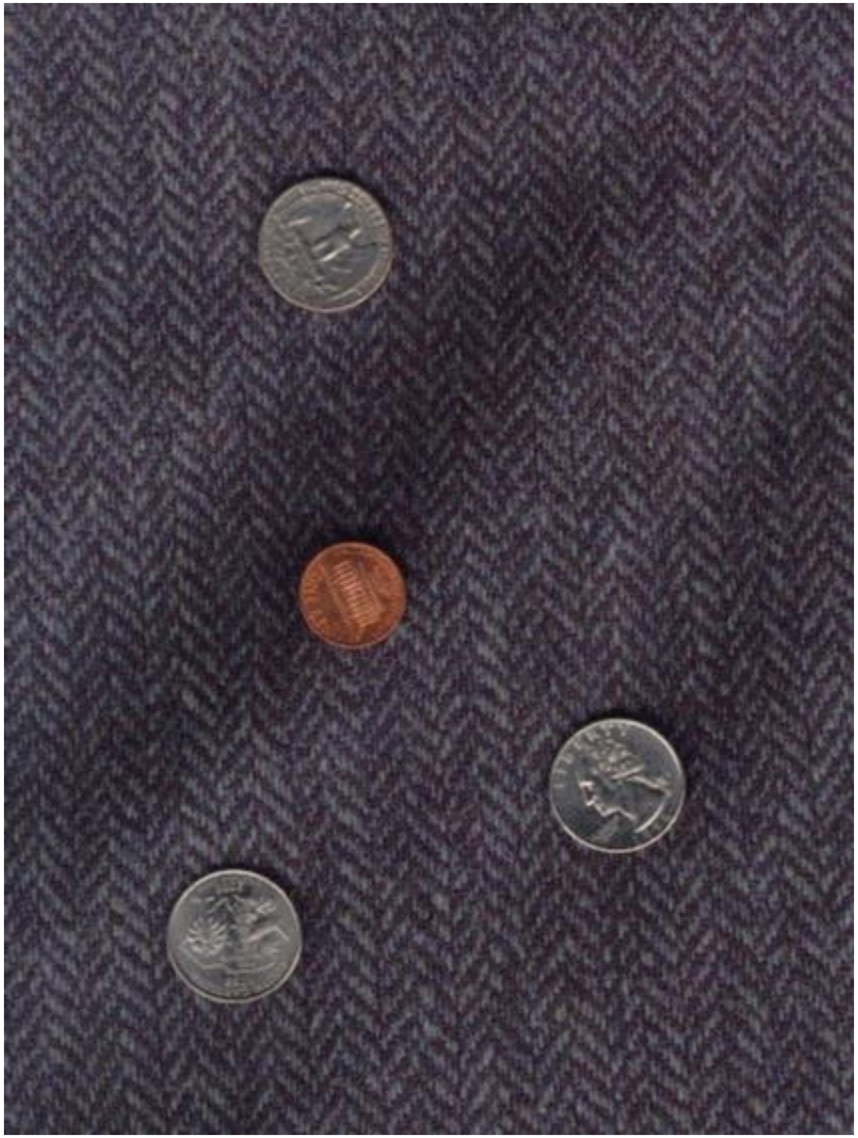
Use Accumulator array $A(a, b, r)$

Real World Circle Examples

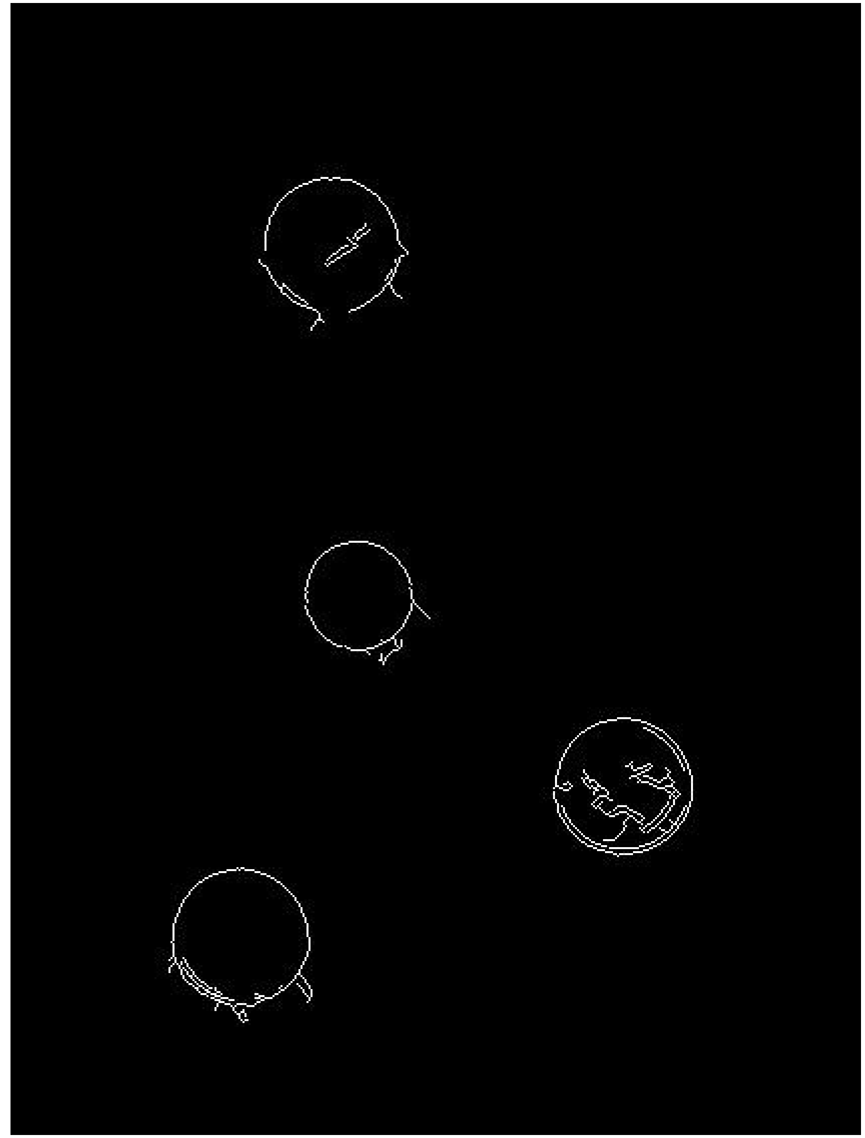
Crosshair indicates results of Hough transform, bounding box found via motion differencing.

Finding Coins

Original

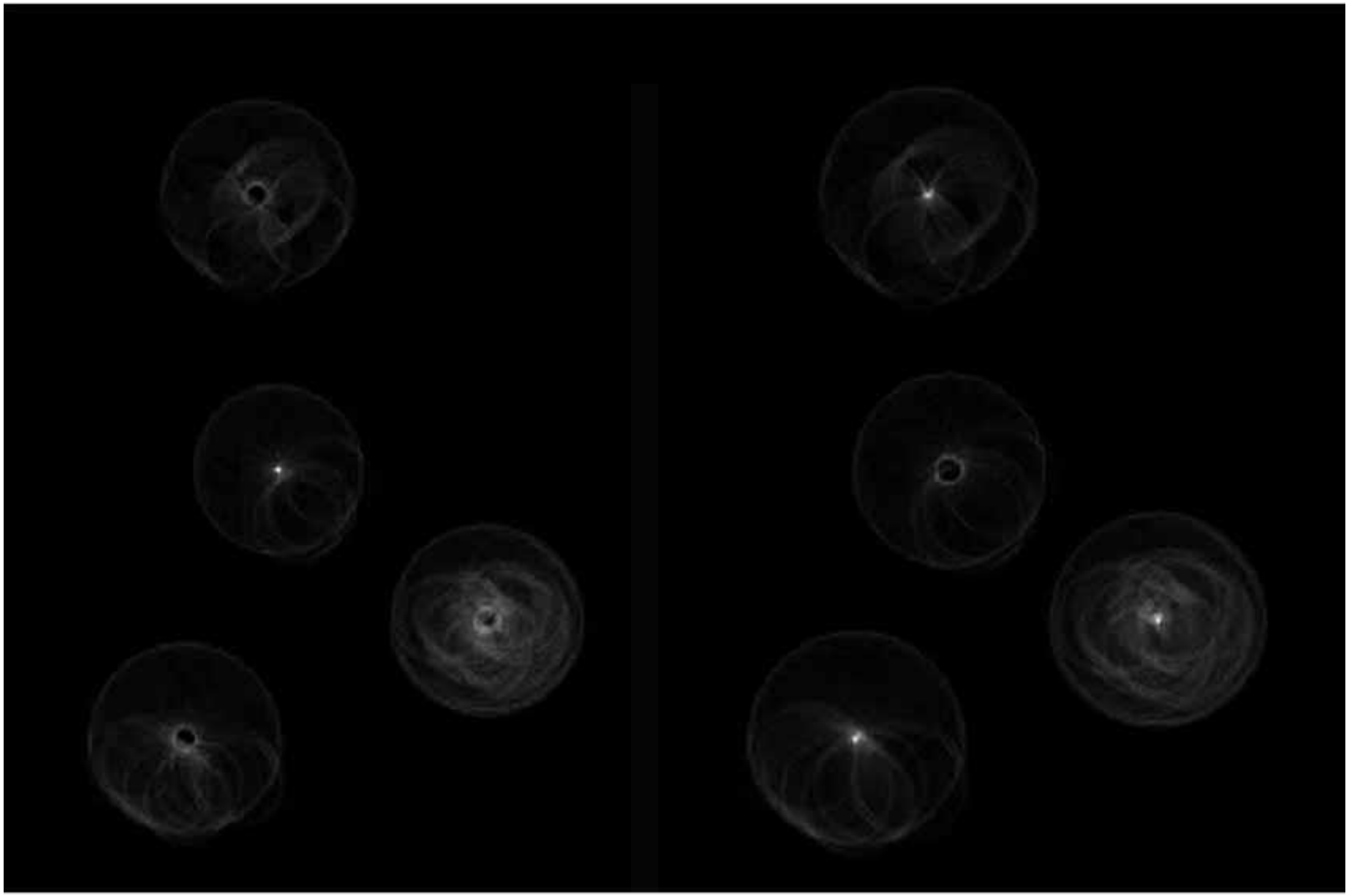


Edges (note noise)

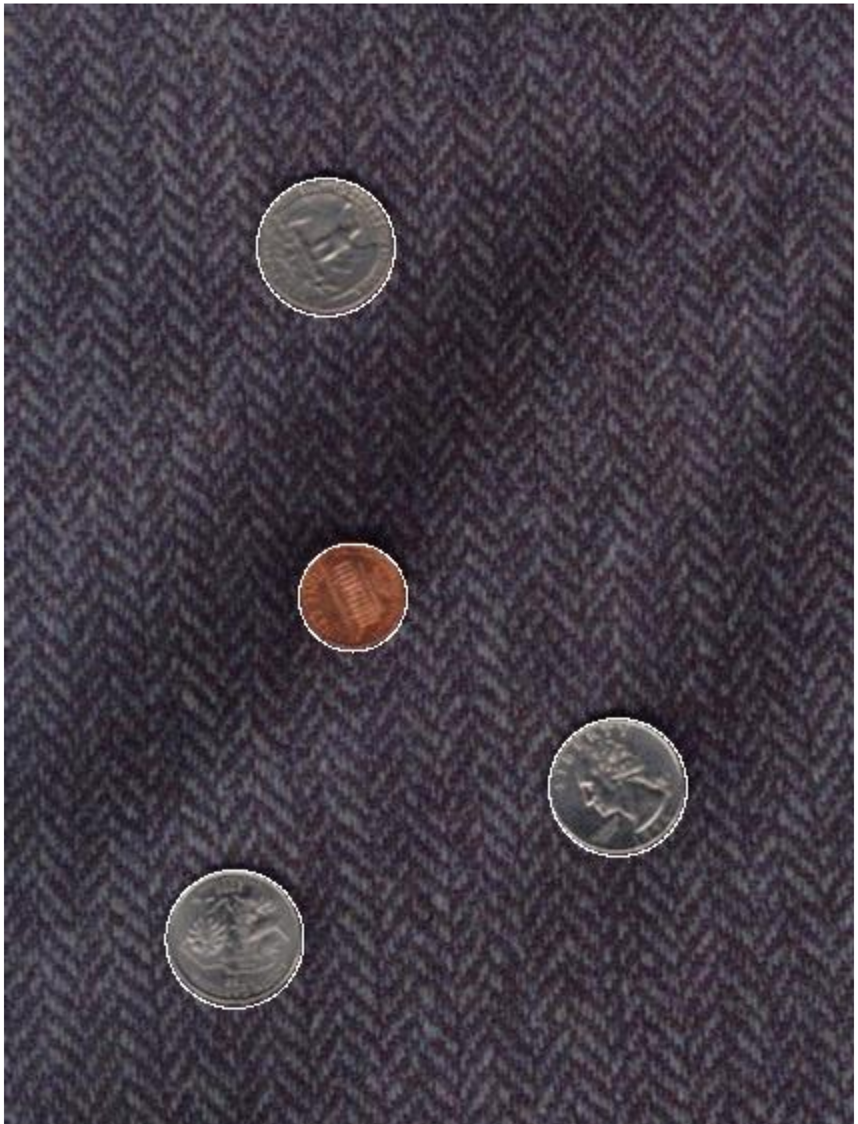


Finding Coins (Continued)

Penn Quarters



Finding Coins (Continued)



Note that because the quarters and penny are different sizes, a different Hough transform (with separate accumulators) was used for each circle size.

Coin finding sample images from: Vivek Kwatra

Hough Transform: Applications

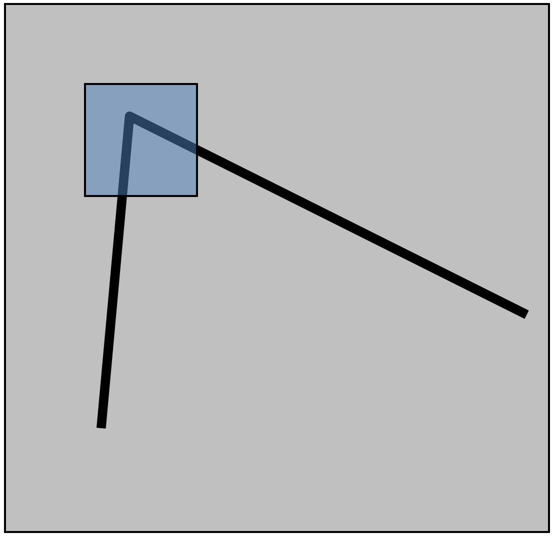
- ❖ Works on Disconnected Edges
- ❖ Effective for simple shapes (lines, circles, Ellipse etc.)
- ❖ Image Recognition
- ❖ Medical field etc.

Corner Detection

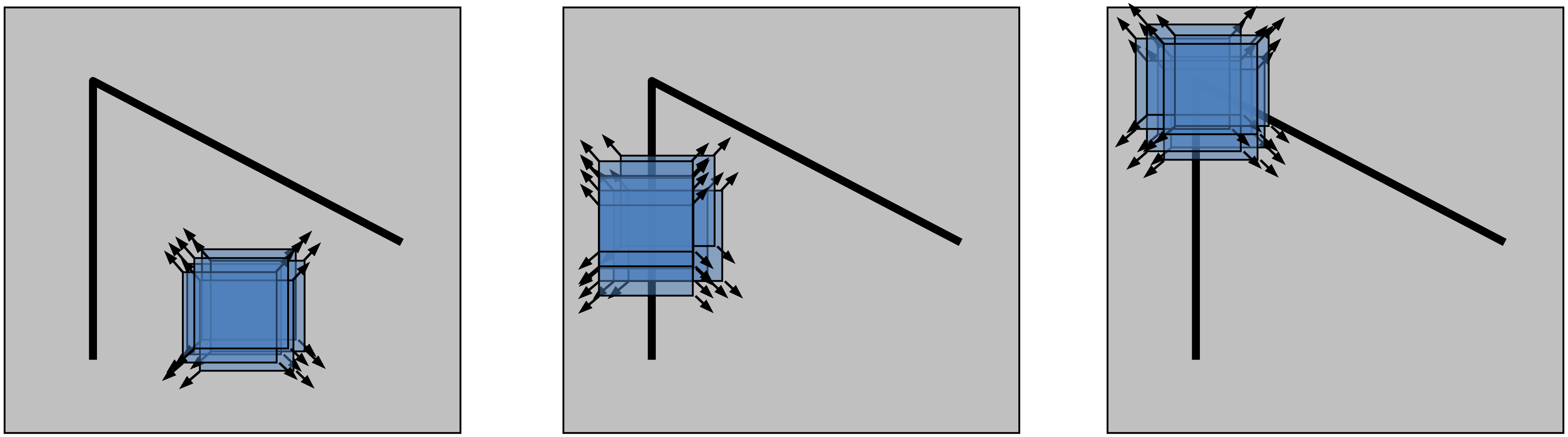
- **A Corner** can be interpreted as **the junction of two edges**, where an edge is a sudden change in image brightness
- A corner is an important feature of an image, which indicates some important points of the image known as **Landmarks Points.**
- It is formed at the boundary when **two bright regions meet**, where the boundary curvature is very high.
- A corner point exhibits the characteristic of having large variations in all directions, unlike edges, where the variation is in only one direction.
- Corner points are also called interest points because the **human visual system quickly identifies landmark points, such as corners, high-contrast regions, and edges.**
- Corners are used in many imaging applications such as **tracking, measurement systems, aligning and stereo imaging.**

Harris Corner Detector

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



“flat” region:
no change in all directions

“edge”:
no change along the edge direction

“corner”:
significant change in all directions

Harris Detector: Mathematics

Change of intensity for the shift [u,v]:

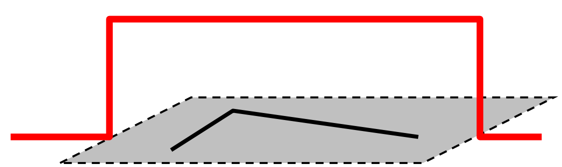
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

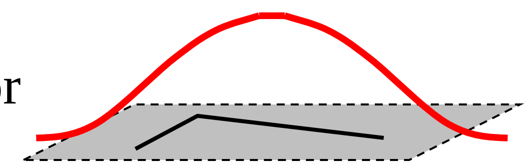
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector

- The autocorrelation function c captures the intensity structure of the neighborhood.
- In addition, it is smoothed by the linear Gaussian-smoothed matrix ***R*** is represented as

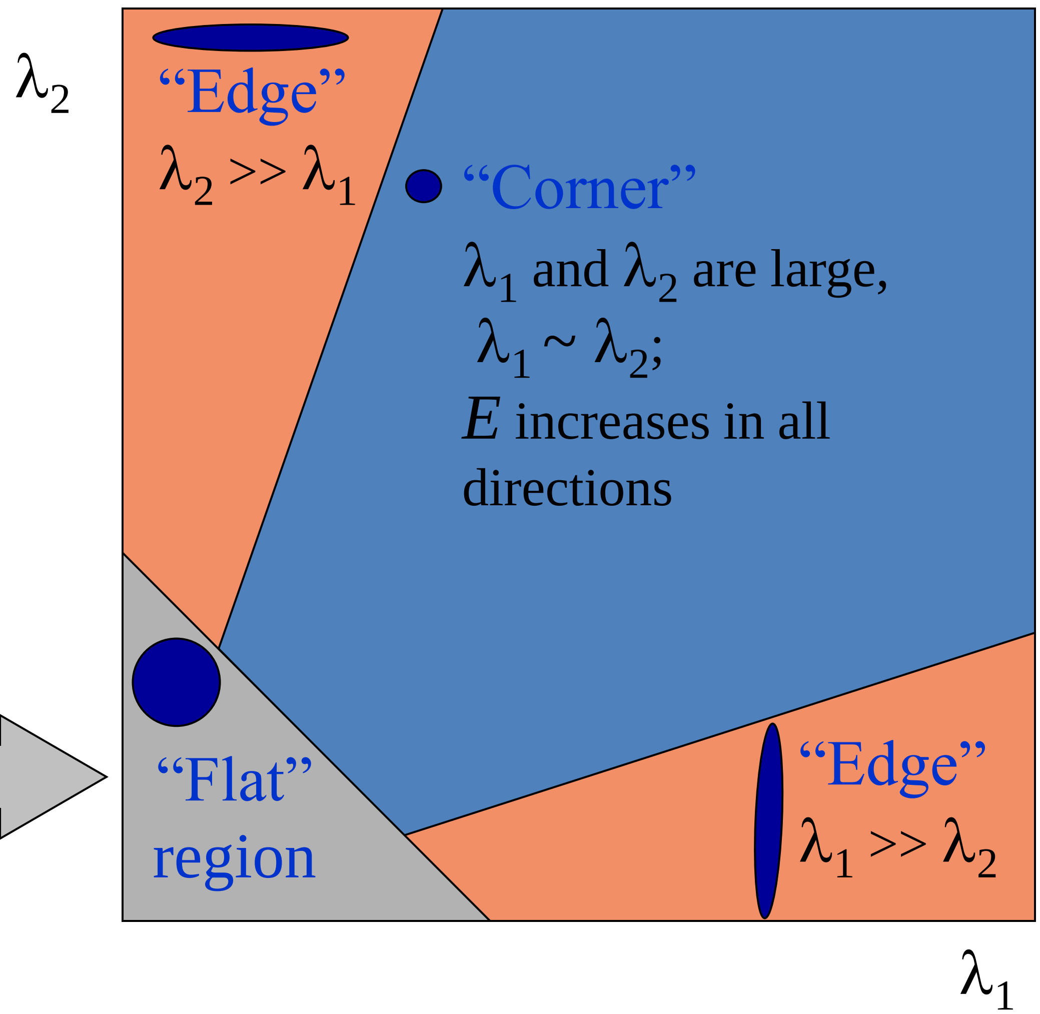
$$R = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

The eigen values of this matrix ***R*** characterize edge strength, and the eigen vectors represent the edge orientation. Assume λ_1 & λ_2 be the eigen values, then

1. If the values of λ_1 & λ_2 are too small, it implies that the window region is flat.
2. If λ_1 is high and λ_2 is small or vice versa, it is an edge.
3. If both λ_1 & λ_2 are high, then it indicates the presence of a corner.

Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:



λ_1 and λ_2 are small;
E is almost constant in all directions

Harris Detector

- In general, the corner strength can be characterized as a corner strength function and represented as A score, Q(u,v), is calculated for each window:

$$Q(u,v)=\det(\boldsymbol{R})-k \times (\mathrm{trace}\ (\boldsymbol{R}^2))$$

The computation of Q(u,v) is easy as the determinant and the trace of the matrix can be expressed in the form of the eigen values are

$$\mathrm{Det}(\boldsymbol{R}) = \prod_{i=1}^{n} \lambda i \quad \text{and} \quad \mathrm{trace}(\boldsymbol{R}) = \sum_{i=1}^{n} \lambda i$$

Harris Corner Detector

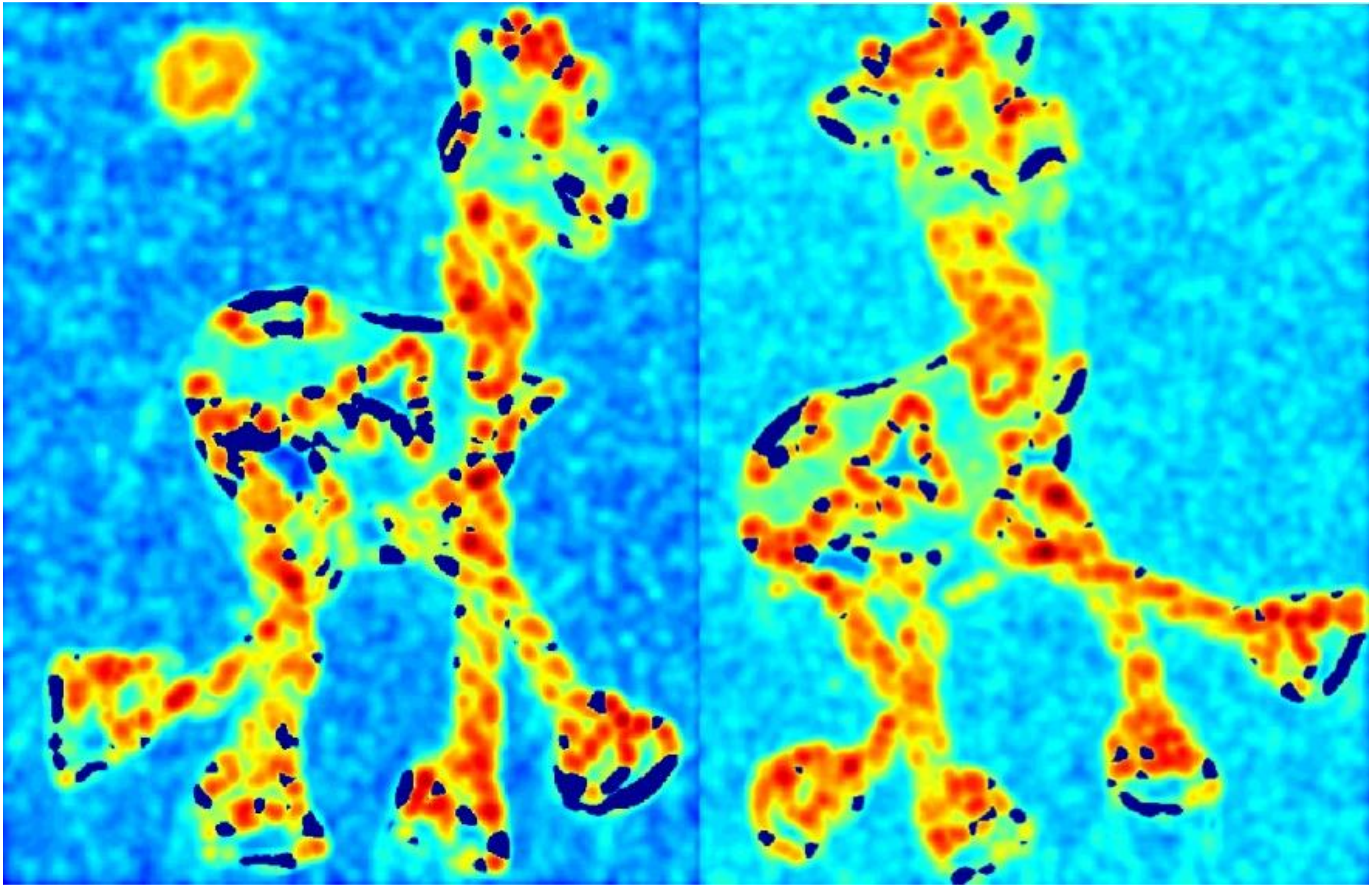
❖ **The Algorithm**:

1. Select an image location (x,y)
2. Identify it is a corner point when strength Q(u,v) > threshold value. Define a window and apply autocorrelation function.
3. Insert into a list of corner points sorted by the corner strength
4. Eliminate the false corners when a weaker corner point lies near a stronger corner.
5. Display all the corner points
6. Exit.

Harris Detector: Workflow

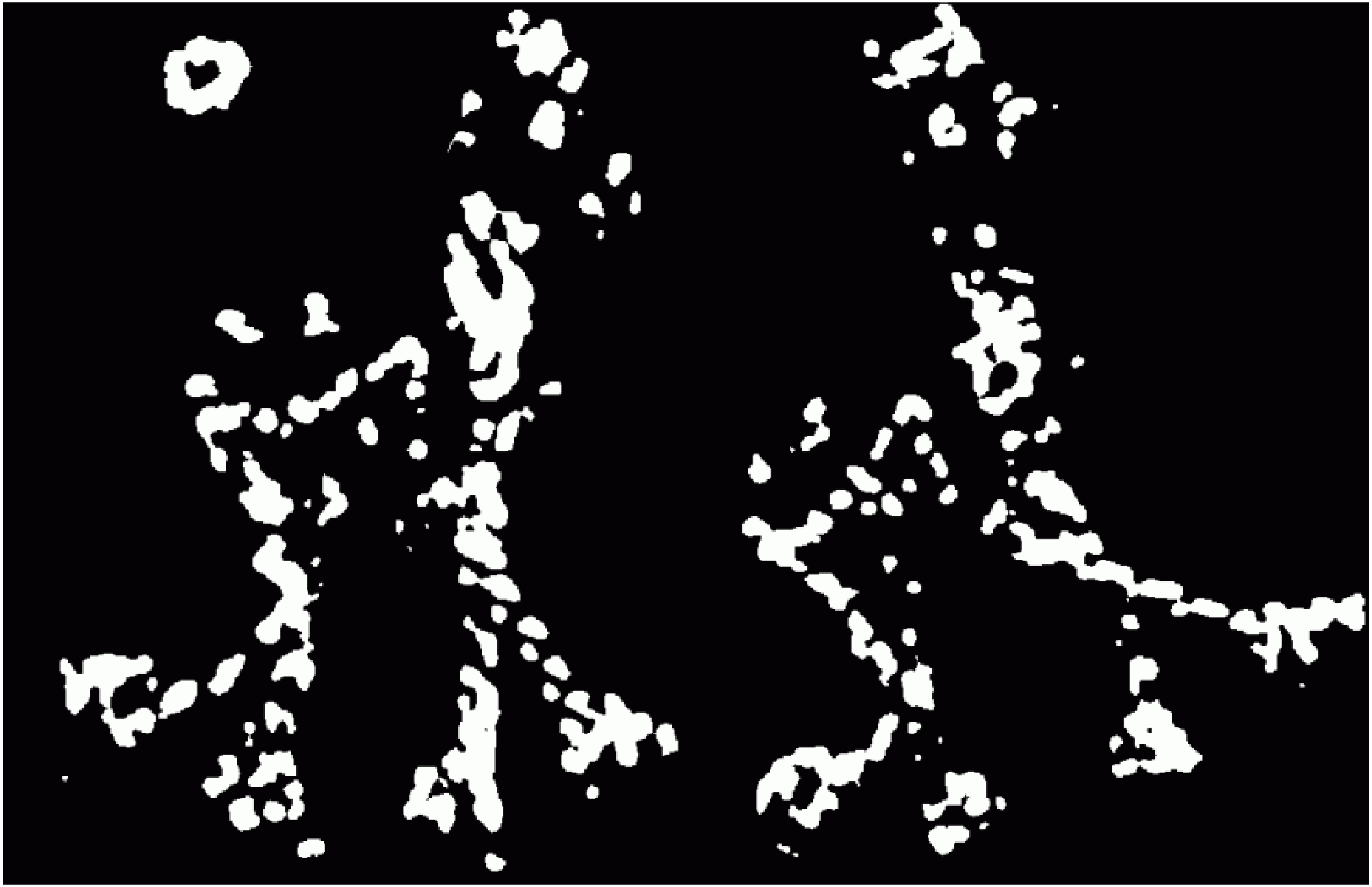
Harris Detector: Workflow

Compute corner response R



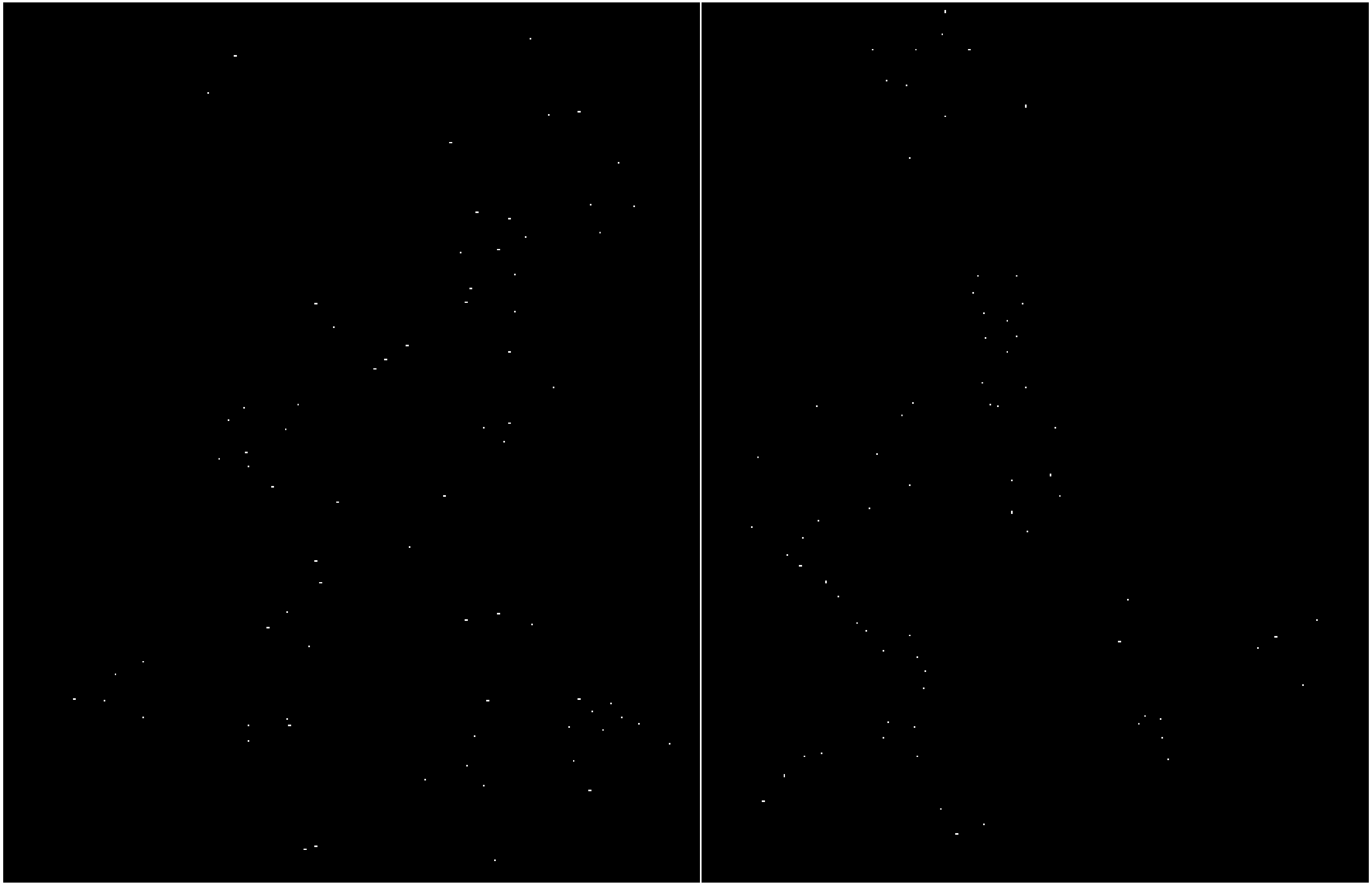
Harris Detector: Workflow

Find points with large corner response: $R>$threshold



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow


ThankYou
Merci
Salamat
Kop
Tack
Hvala
Gra
Xie
kun
Diolch
Kiitos
Sheun
umesc
Kasih
Mamnoon
Todah
Shnorhakalutiun
Gamsahapnida
Shokriya
Ngiyabonga
Dziękuję
Shokrun
Spaas
Mul
Ači
Dank
Takk
Te
ekkür
Dekuju/Dekujeme
Cam
or
al
Dhanyavaad
Dakujem
Waad
Daw
Dhanyavaadaalu
Dhanyavad
Khopjai
Dankie
Go
Grazie
Faleminderit
Kruthagnathalu
krap
Arigatou
Dhonnobaad
Shukriya
Or
ederim
Asante
Hain
Dhan
daa
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun

Hough Transform & Harris Corner Detection

By,

Prof(Dr.) Premanand P Ghadekar

Outline

- Edge Linking and Boundary Detection
- Introduction to Hough Transform
- Straight line fitting
- Hough Transform Algorithm
- Updated Hough Transform Algorithm
- Harris Corner Detection

Image Feature Extraction-Recapitulation

- ❖ Edges (edge pixels)
- ❖ Sobel, Roberts, Prewitt
- ❖ Laplacian of Gaussian
- ❖ Canny Edge detector
- ❖ Watershed Segmentation
- ❖ Region Growing
- ❖ Split & Merge Algorithm etc.

Shape Features

- Straight Lines
- Circles and Ellipses
- Arbitrary Shape

Edge Linking and Boundary Detection

❖ Edge Linking Procedures assemble edge points into meaningful boundaries.

Edge Linking

Local Processing **Global Processing**

Local Processing

- Take an Edge detected Image
- Analyze each pixel in a small neighborhood of every point (x,y) .
  - All points that are similar in nature are linked
  - This forms a Boundary of pixels that are similar in nature.

- Similarity
  - Strength of the response of gradient operator
  - Direction of the gradient

Local Processing

- Edge pixels (x,y) & (x', y') are similar if

$$|\nabla f(x,y) - \nabla f(x',y')| \leq T$$

$\nabla f(x,y)$-Gradient at x,y & $\nabla f(x',y')$-Gradient at x',y'

$$|\alpha(x,y) - \alpha(x',y')| \leq A$$

$\alpha(x,y)$- Direction at (x,y) & $\alpha(x',y')$ –Direction at (x',y')

$$(x',y') \in N_{xy}$$

T is non-negative threshold & A is Angle threshold

Global Processing-Hough Transform

- If (x’, y’) is **not with in the neighborhood** of (x,y), then (x’,y’) can not be linked to (x,y). **Local processing technique does not help.**
- **Global Processing technique –Hough Transform** can be used
- Method and Means for Recognizing Complex Patterns, Paul V. C. Hough et al (Patent)
- Hough Transform is the **Mapping from Spatial domain to Parameter space**

Hough Transform

❖ **Elegant method for Direct Object Recognition**

➢ The **Hough transform** is a feature extraction technique used in image analysis, computer vision, and digital image processing.

➢ The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by **a Voting Procedure.**

➢ This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called **Accumulator Space** that is explicitly constructed by the algorithm for computing the Hough transform.

➢ The classical Hough transform was concerned with the **identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses.**

Hough Transform

- The Hough transform takes the images created by the edge detection operators.
- Most of the time, the edge map generated by the edge detection algorithms is disconnected.
- The Hough transform can be used **to connect the disjointed edge points**.
- Unlike the edge linking algorithms, the Hough transform does not require any prerequisites to connect the edge points.
- It is used **to fit the points as plane curves.**
- The plane curves typically **are lines, circles, and parabolas.**
- The equation of a line is given below

$$y=mx+c$$

Where **m** is the slope of the line, **c** is **y** intercept of the line.

Image and Parameter Spaces

Equation of Line: $y = mx + c$

Find: (m, c)

Consider point: (x_i, y_i)

$$y_i = mx_i + c \quad or \quad c = -x_i m + y_i$$

Parameter space also called **Hough Space**

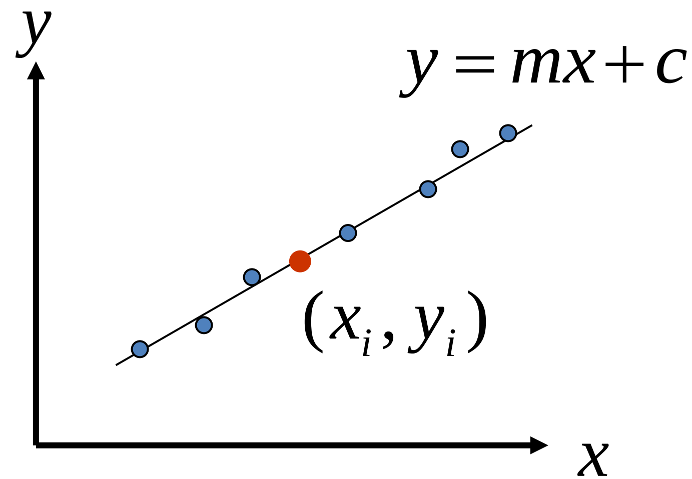
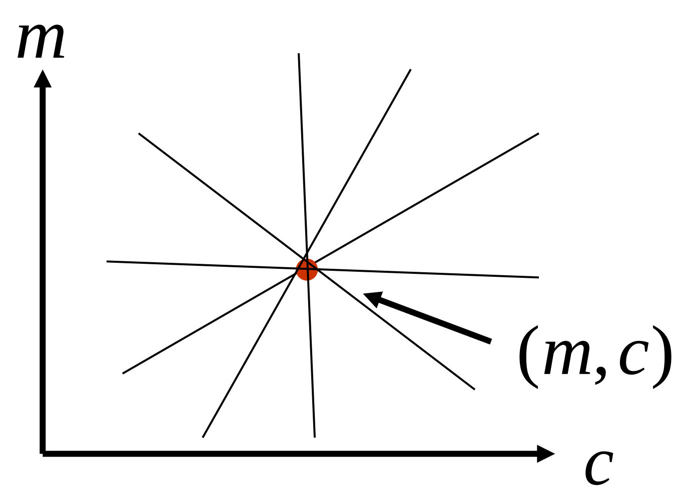


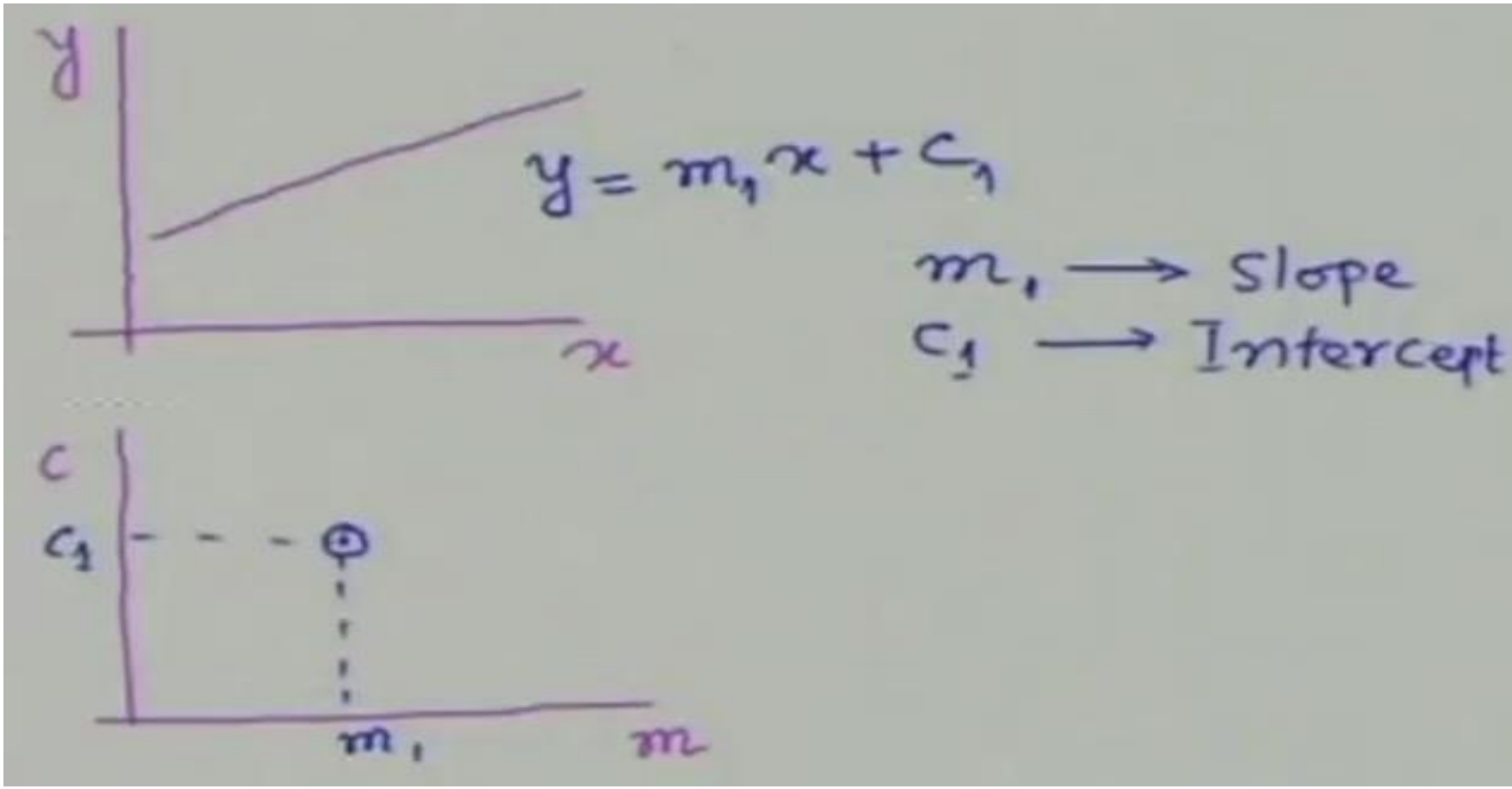
Image Space



Parameter Space

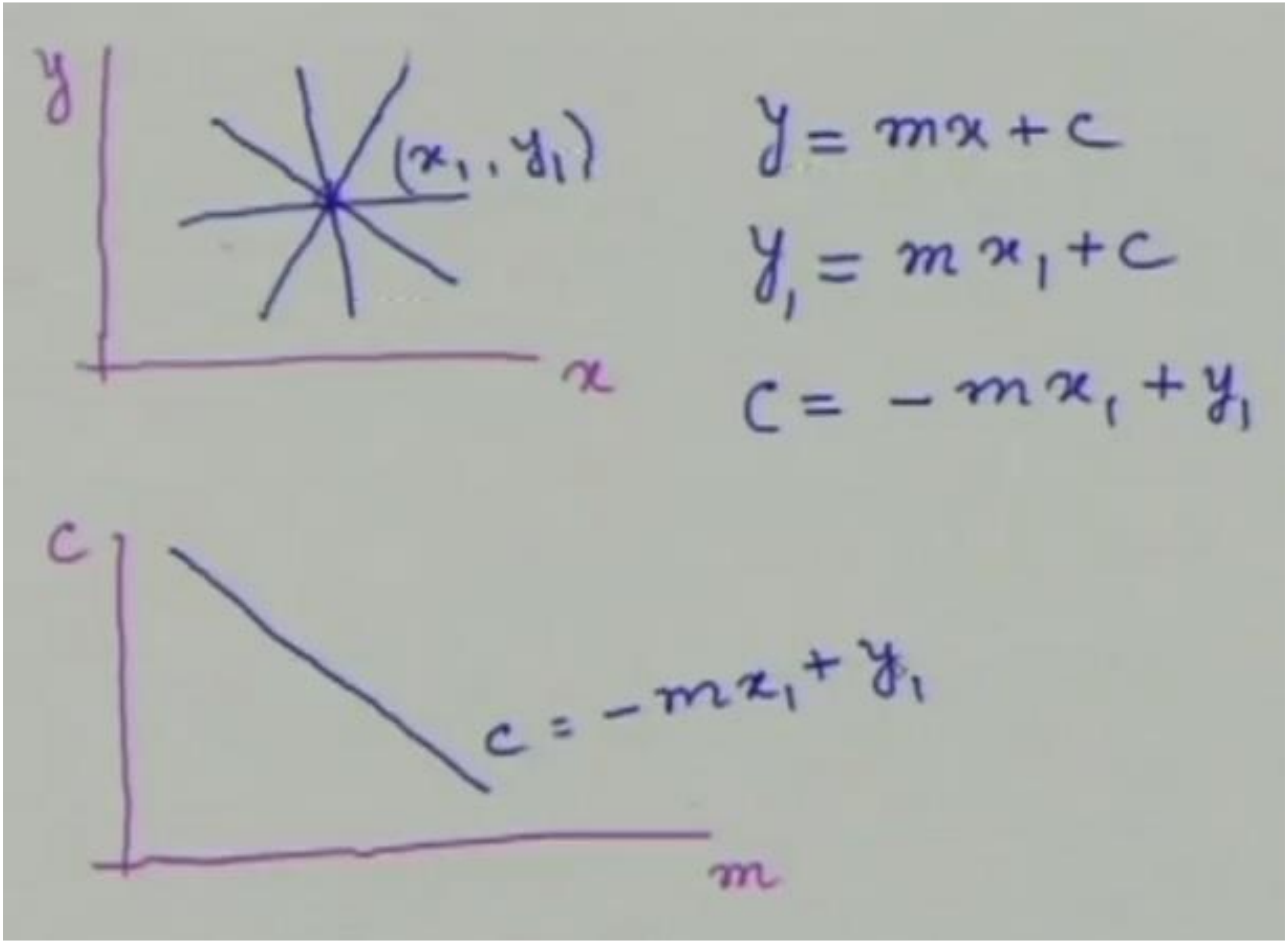
Image and Parameter Spaces-Line

An edge point in x-y plane is transformed to a c-m plane. Line eq- **y=mx+c**
Write the line equation as **c=-mx+y**



A set of **points** are said to be **collinear** if they all lie on a single line. Because there is a line between any two **points**, every pair of **points** is **collinear**.

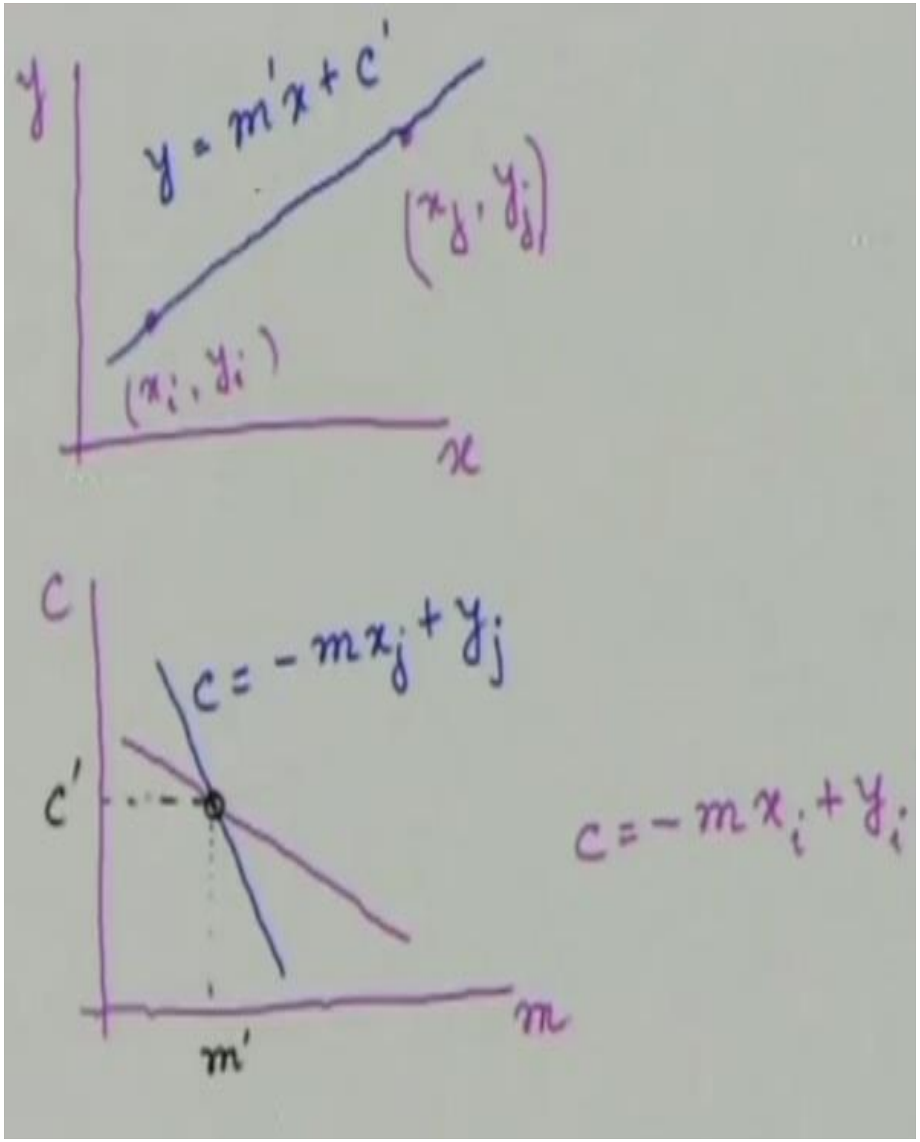
Image and Parameter Spaces-Point



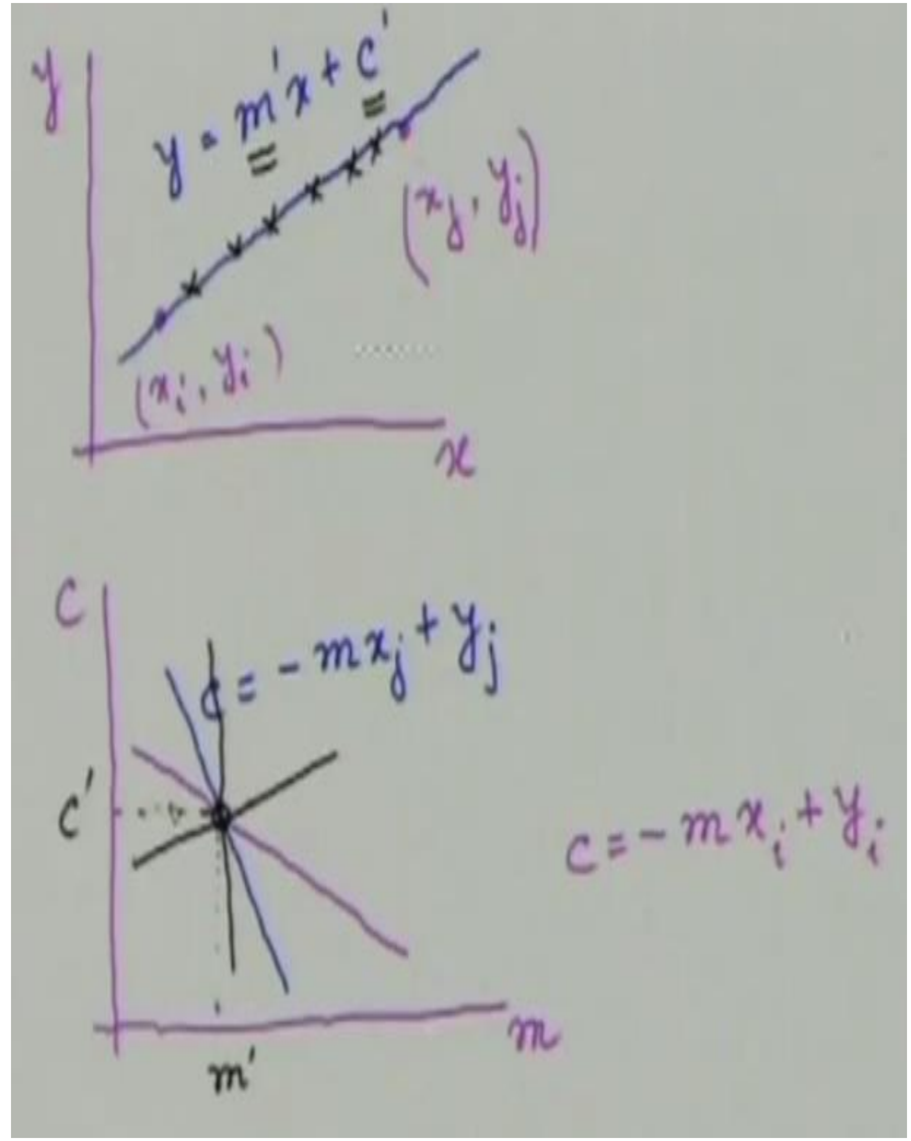
Hough Transform

- All the edge points (xi,yi), (xi+1,y_{i+1})…….(x_j,y_j) in the x-y plane need to be fitted .
- Hence the x-y plane should be transformed into a different c-m plane.
- All points are lines in the c-m plane.
- The objectives is to find the intersection point.
- A common intersection point indicates that the edge points are part of the same line.
- If A & B are points connected by a line in the spatial domain, then they will be intersecting lines in the Hough Space.

Image and Parameter Spaces-Two Points

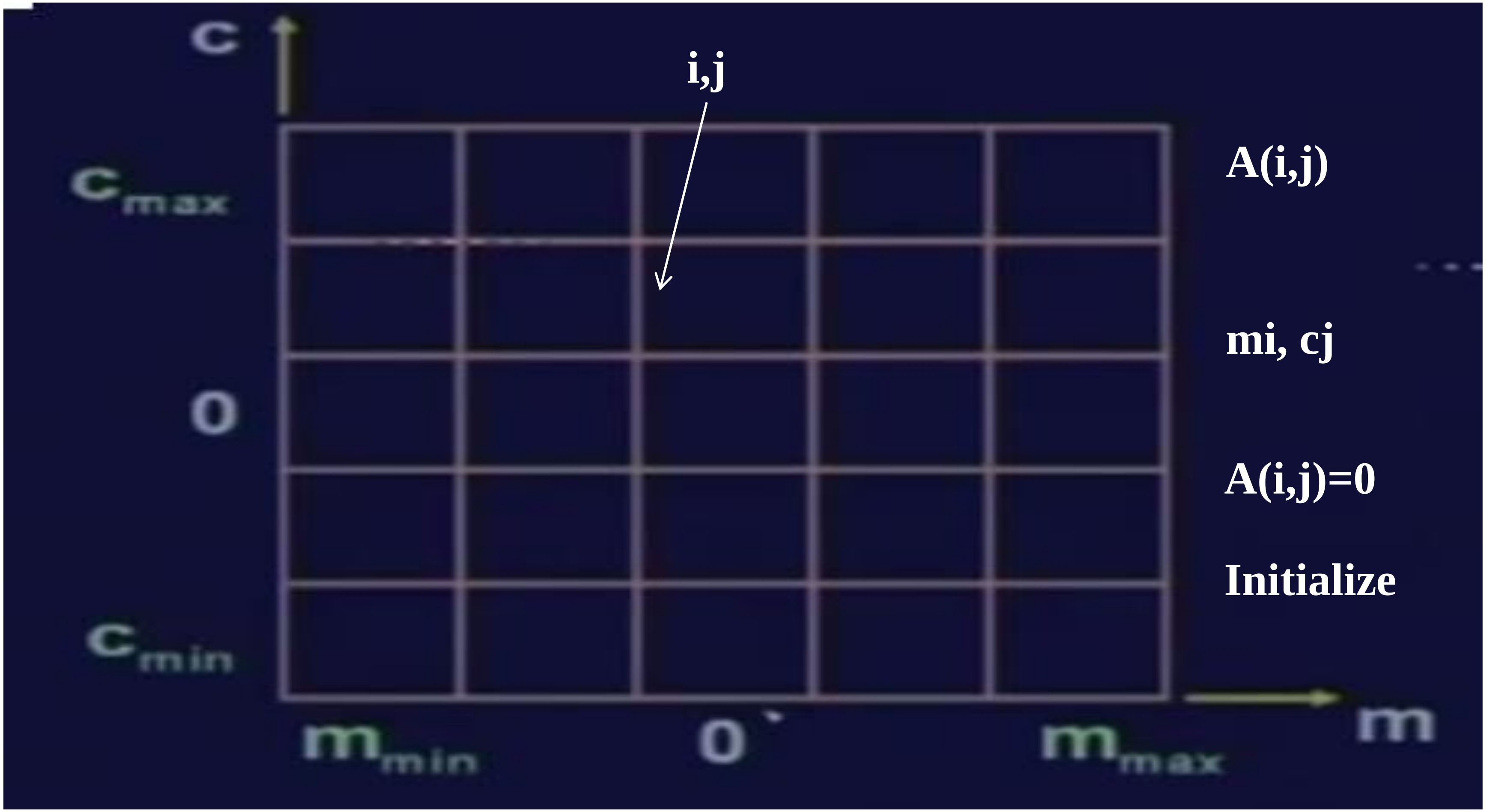


(a)



(b)

Accumulator Array-Digital –c-m plane



Where- m_{min} is minimum Slope value & m_{max} is the maximum slope value. C_{min} is the minimun intercept value of c and C_{max} is the maximum intercept value of c

Hough Transform Algorithm

❖ **The Hough Transform is stated as follow**

1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image :-

   If the pixel (x_k , y_k) is an edge pixel, then

(a) $c=(-m)x_k +y_k$

(b) $A(p, q)=A(p, q)+1$

5. Show the Hough Space.

   $A(i,j)=Q$

6. Find the local maxima in the parameter space
7. Draw the line using the local maxima.

Accumulator Array-Digital –c-m plane

Step 1. Quantize parameter space (m, c)

Step 2. Create accumulator array $A(m, c)$

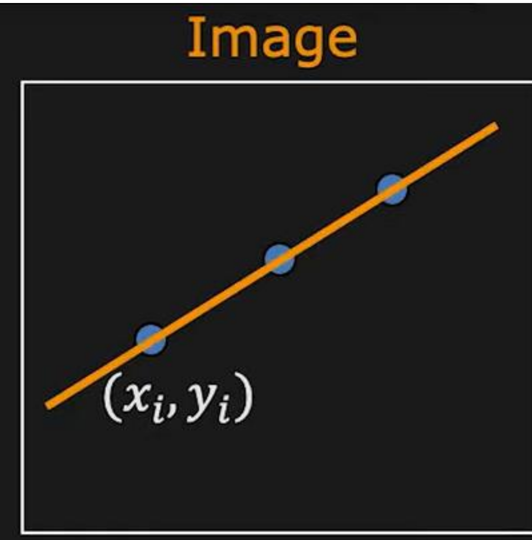
Step 3. Set $A(m, c) = 0$ for all (m, c)

Step 4. For each edge point (x_i, y_i),

$$A(m, c) = A(m, c) + 1$$

if (m, c) lies on the line: $c = -mx_i + y_i$

Step 5. Find local maxima in $A(m, c)$

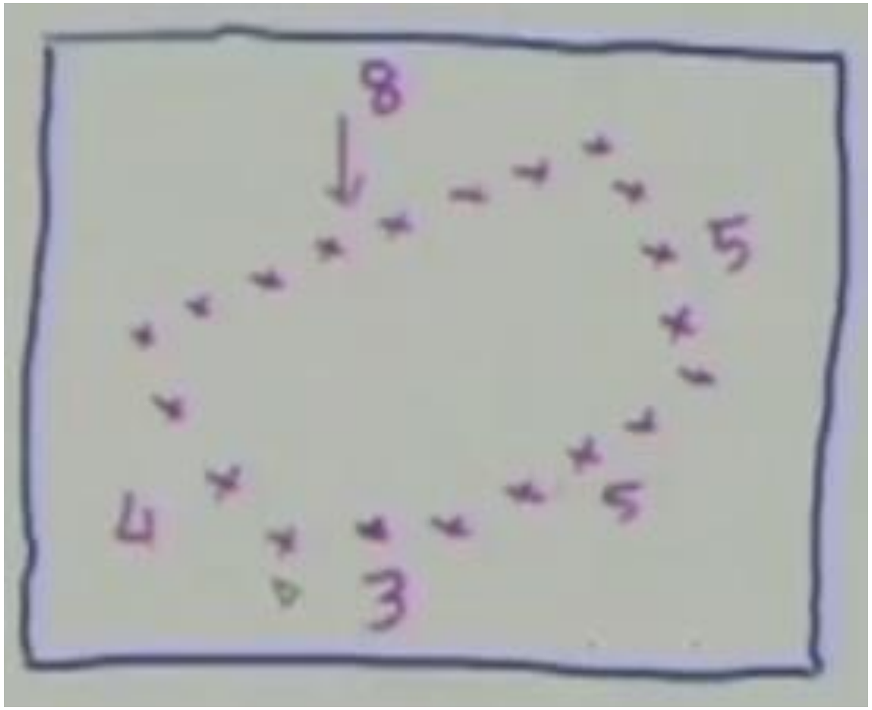


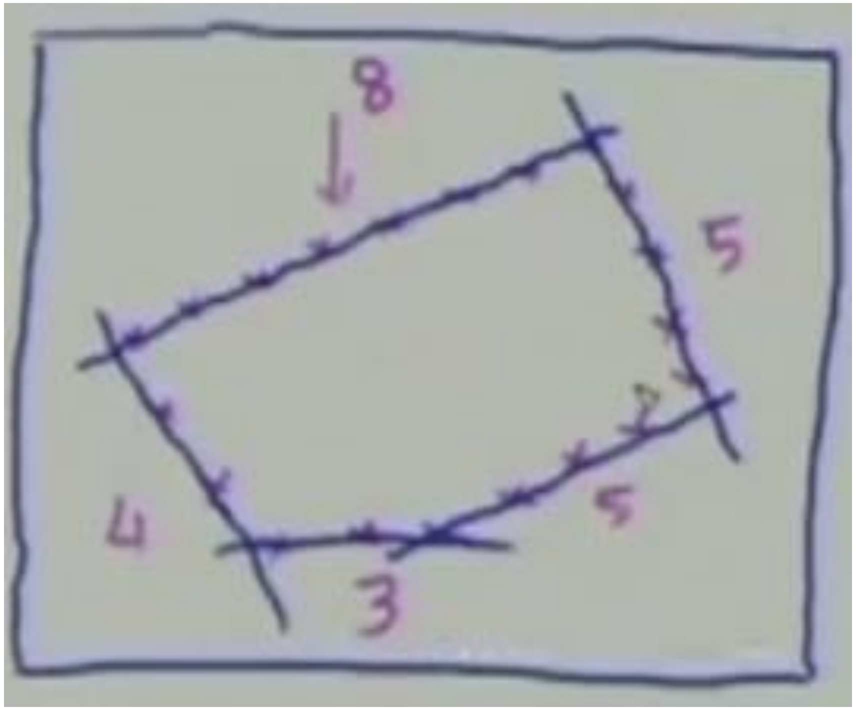
$A(m, c)$

c

1	0	0	0	1
0	1	0	1	0
1	1	3	1	1
0	1	0	1	0
1	0	0	0	1

Hough Transform-Line Detection Example





- ❖ **Problem-It does not work for vertical lines, as they have a slope of infinity. Therefore, it is better to convert this line in to polar coordinates.**

Modified Hough Transform

- ❖ The line in polar coordinates form can be represented as $\rho = x \cos\Theta + y \sin\Theta$, where Θ is the angle between the line and x-axis, and ρ is the diameter.
- ❖ **The Modified Hough Transform-**

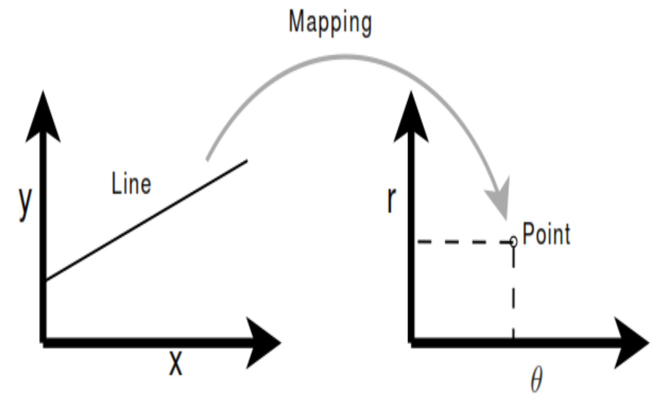
1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image:-

   If the pixel is an edge pixel, then for all Θ

   (a) Calculate ρ for the pixel, then for all Θ

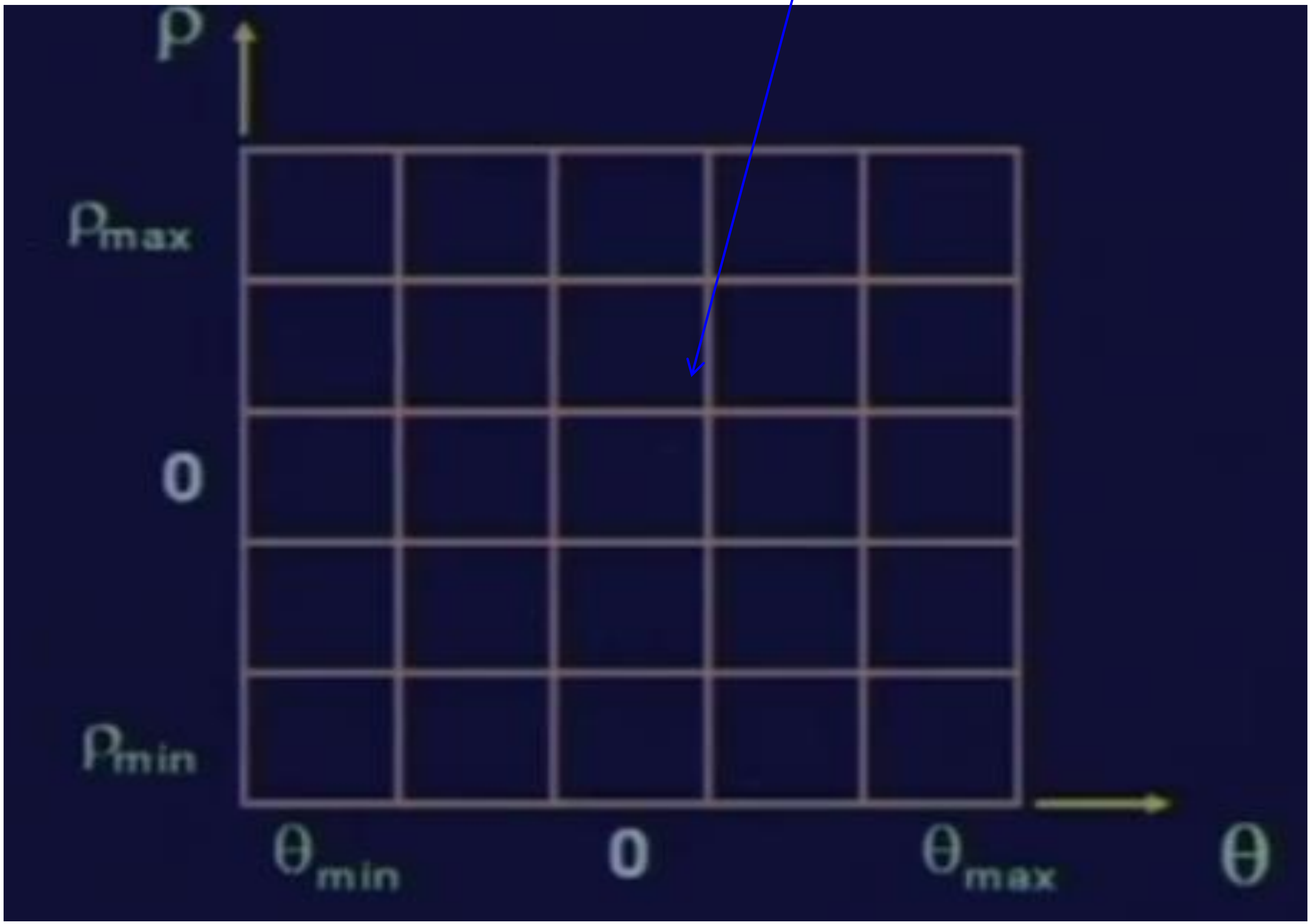
   (b) Increment the position (ρ,Θ) in the accumulator array *P*.

5. Show the Hough Space.
6. Find the local maxima in the parameter space.
7. Draw the line using the local maxima.



Modified Hough Transform

(i,j)

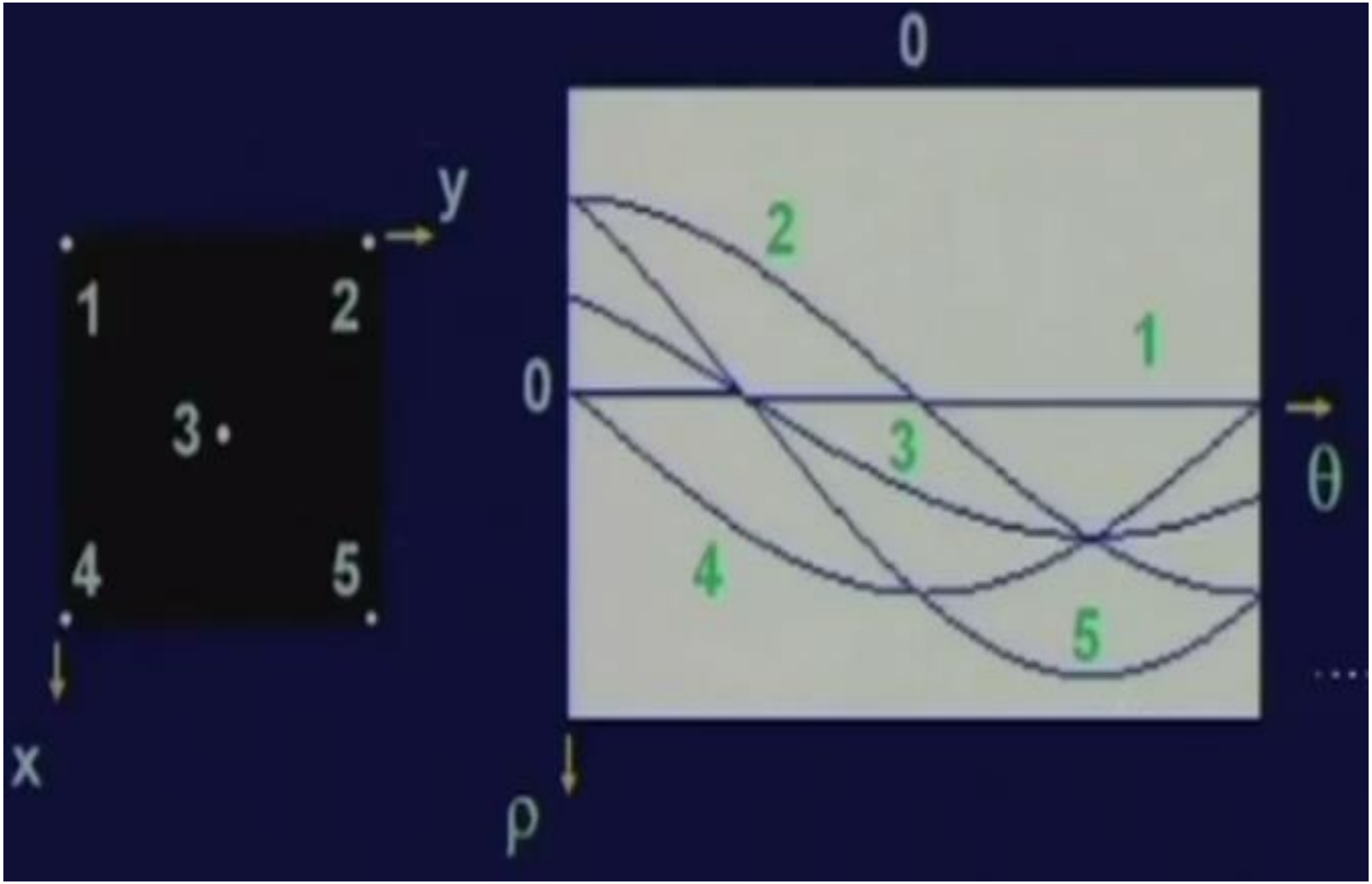


A(i,j)

θi, ρj

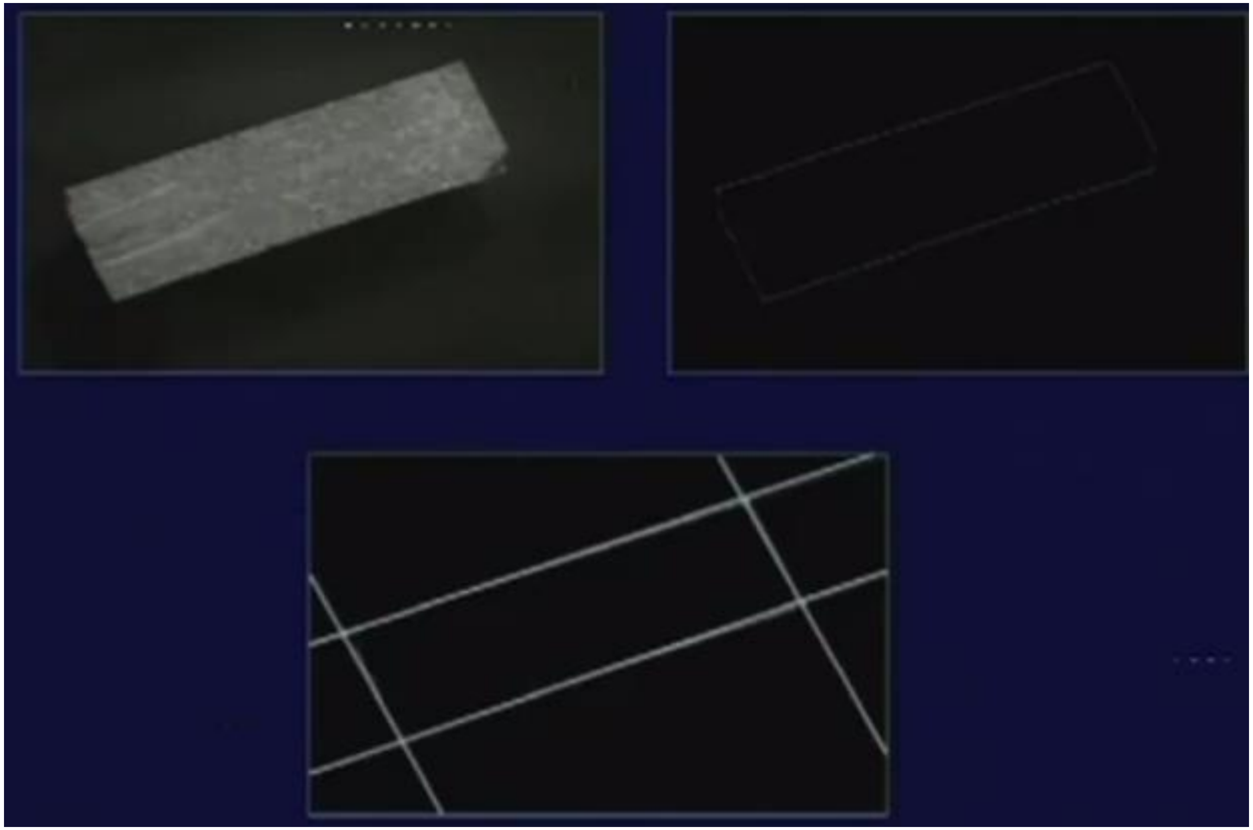
$\rho = X_k \cos\theta + Y_k \sin\theta$

Modified Hough Transform



Hough Transform

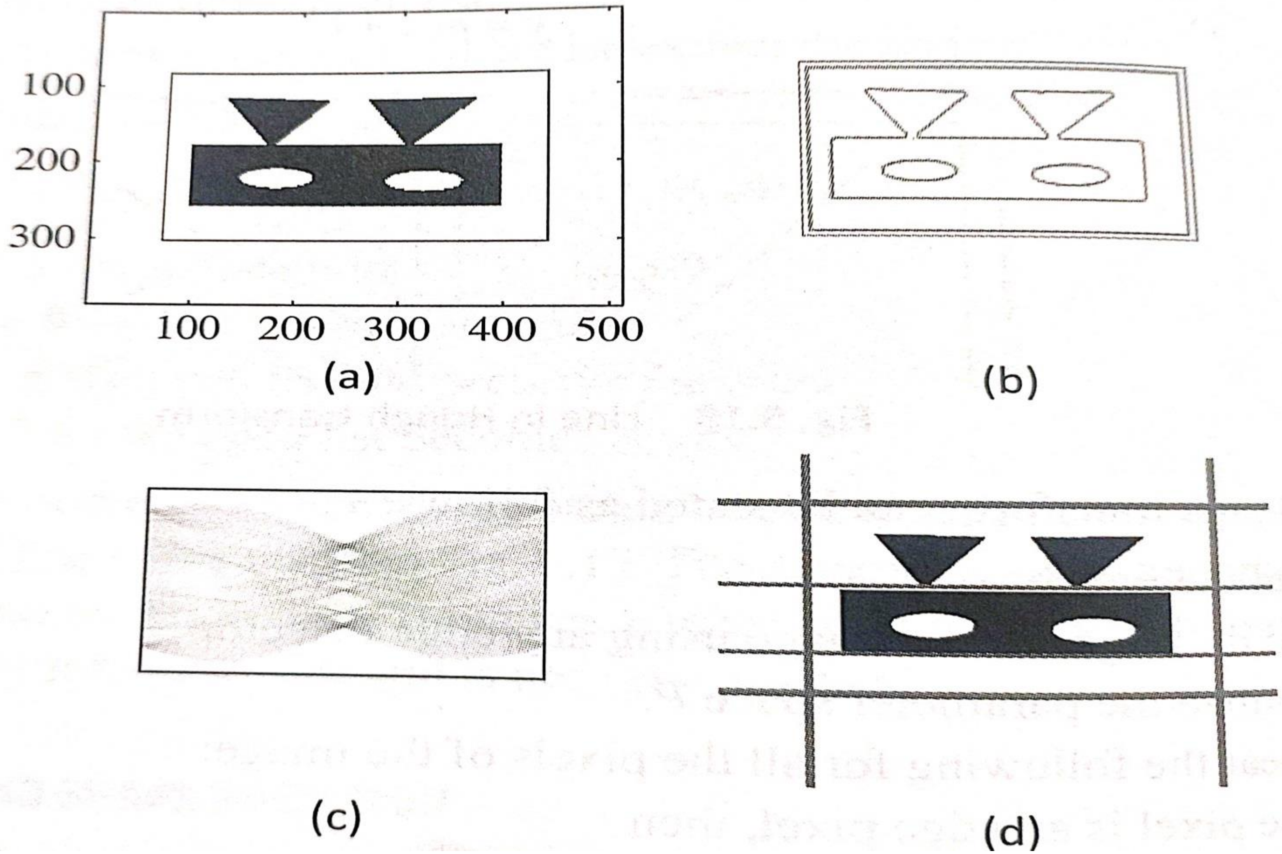
❖ **The result of the Hough Transform line detection is shown below**



(a) Input image (b) Output Edges (c) Detected Lines

Hough Transform

❖ **The result of the Hough Transform line detection is shown below**



(a) Input image (b) Output Edges (C) Accumulator Image (D) Detected Lines

Hough Transform-Circle Equation

A circle with center at (3,4) and a radius of 6:

Soln- Circle Equation-

$$(x-a)^2 + (y-b)^2 = r^2$$

Put in (a, b) and r:

$$(x-3)^2 + (y-4)^2 = 6^2$$

Hough Transform for Circle Detection

- The Hough Transform for other shapes can also be found.
- The Hough Transform for a **circle detection** can be given as follows:-

$$(x-a)^2 + (y-b)^2 - r^2 = 0$$

The parameter space is three dimensional as the unknown are a, b, and r.

The polar form Circle detection algorithm is-

1. Load the image
2. Find the edges of the image using any edge detector.
3. Quantize the parameter space *P*.
4. Repeat the following for all the pixels of the image:-

   If the pixel is an edge pixel, then for all values of r, Calculate

(a) $a = x - r\cos\Theta$

(b) $b = y - r\sin\Theta$

(c) $A(a, b, r) = A(a, b, r) + 1$

5. Show the Hough Space.
6. Find the local Maxima in the parameter space.
7. Draw the Circle using the local Maxima.

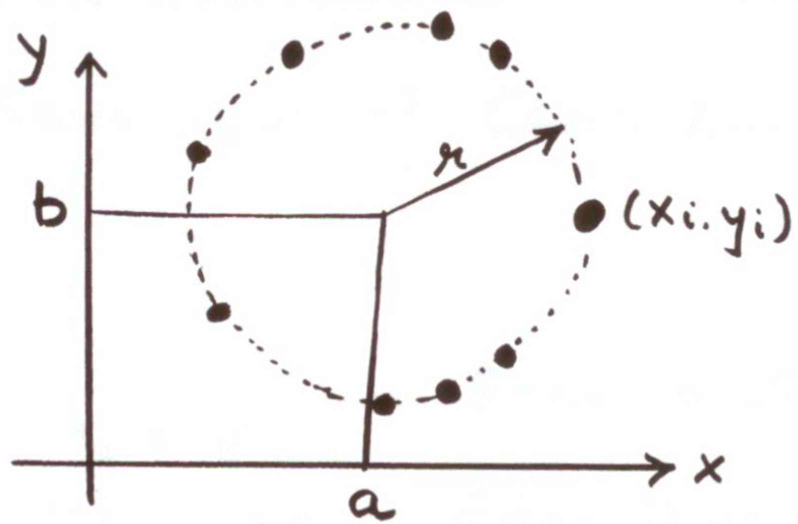
Finding Circles by Hough Transform

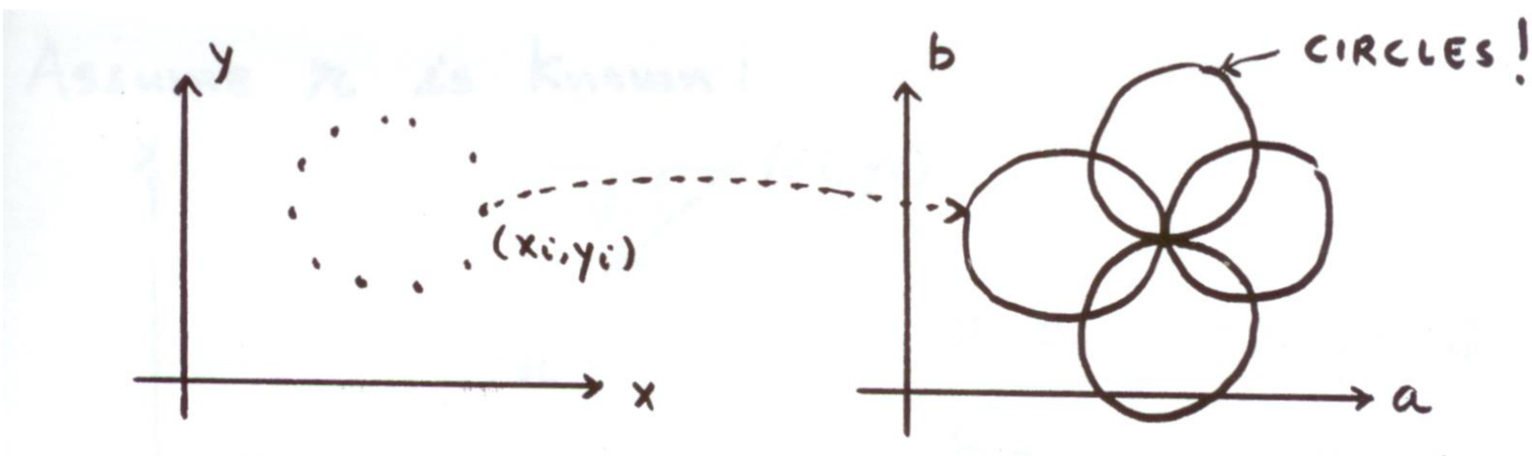
Equation of Circle:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

If radius is known: (2D Hough Space)

Accumulator Array $A(a,b)$

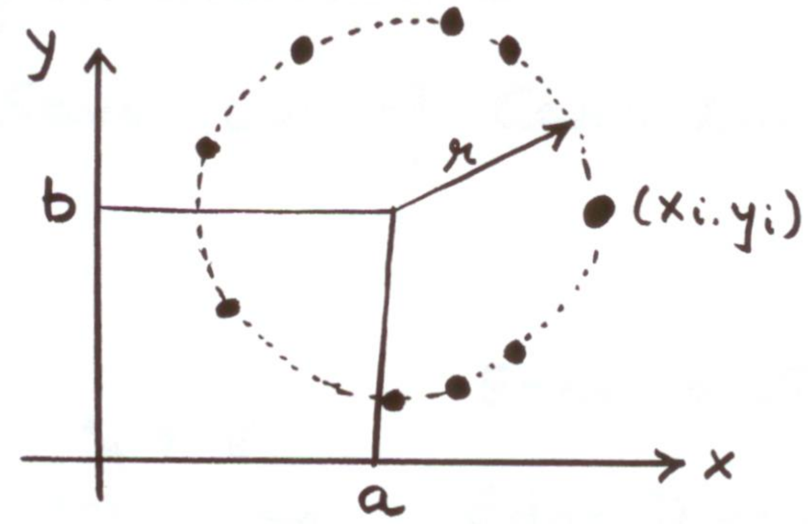




Finding Circles by Hough Transform

Equation of Circle:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$



If radius is not known: 3D Hough Space!

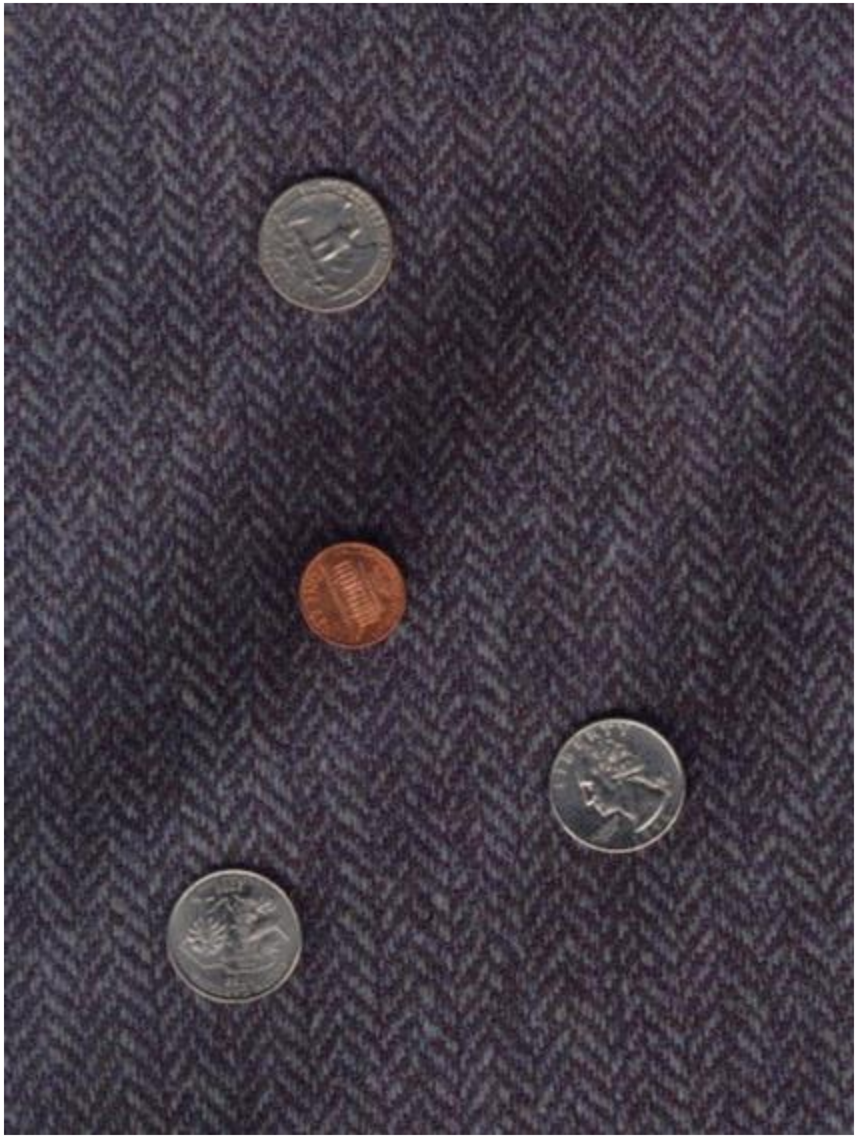
Use Accumulator array $A(a, b, r)$

Real World Circle Examples

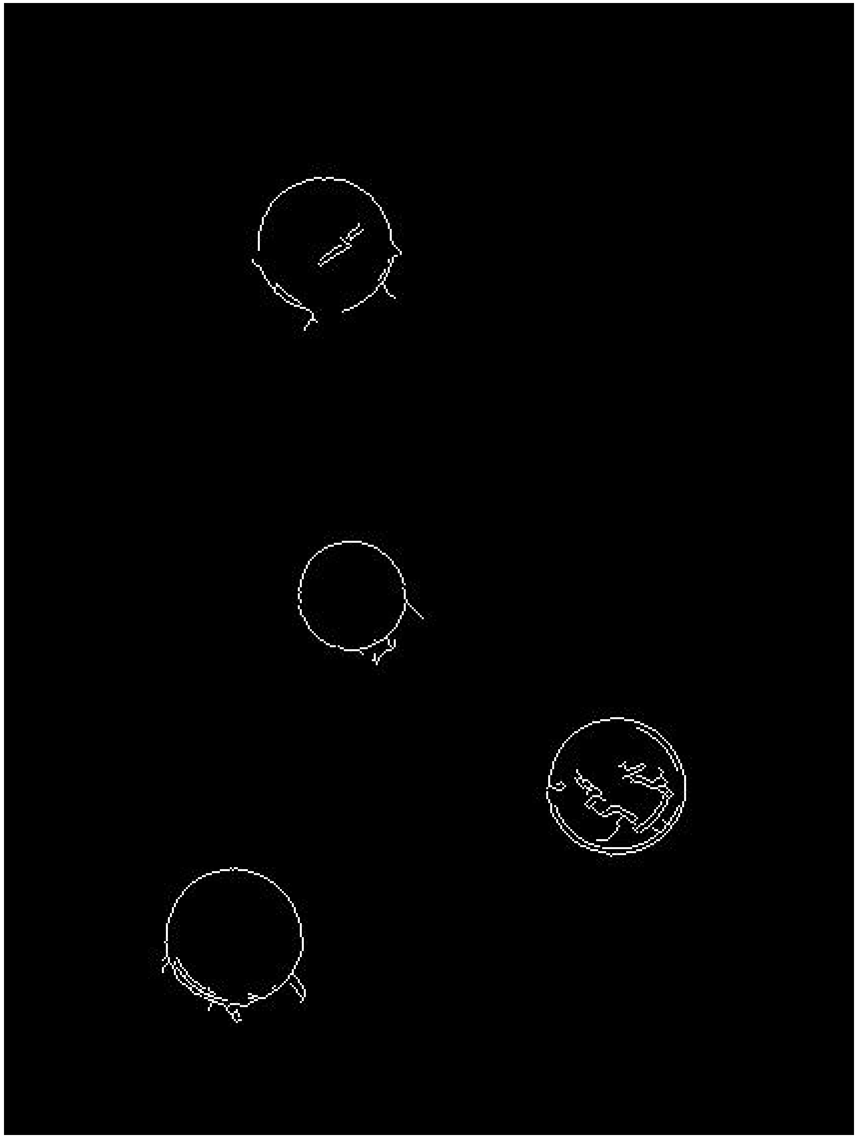
Crosshair indicates results of Hough transform, bounding box found via motion differencing.

Finding Coins

Original

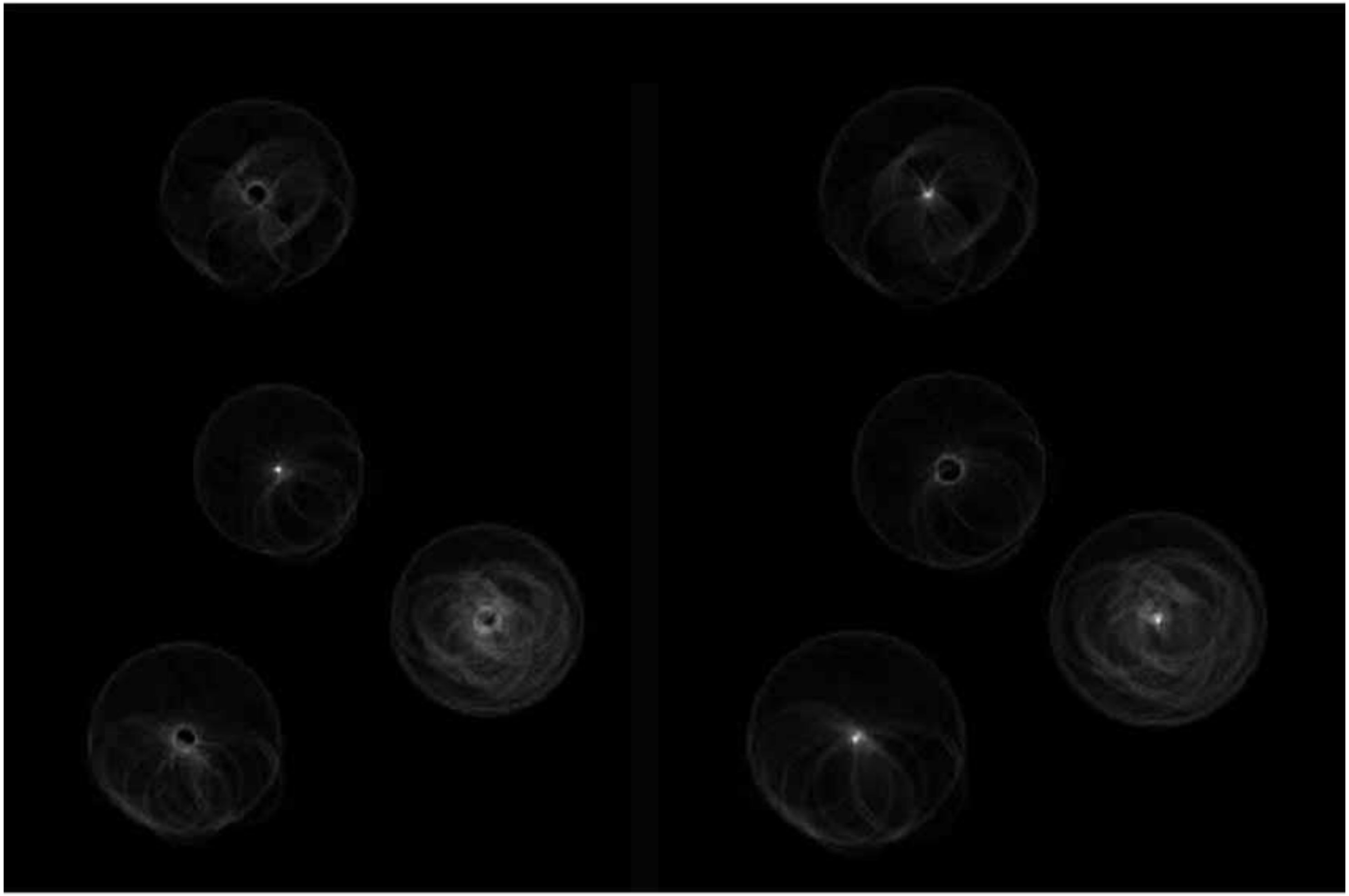


Edges (note noise)

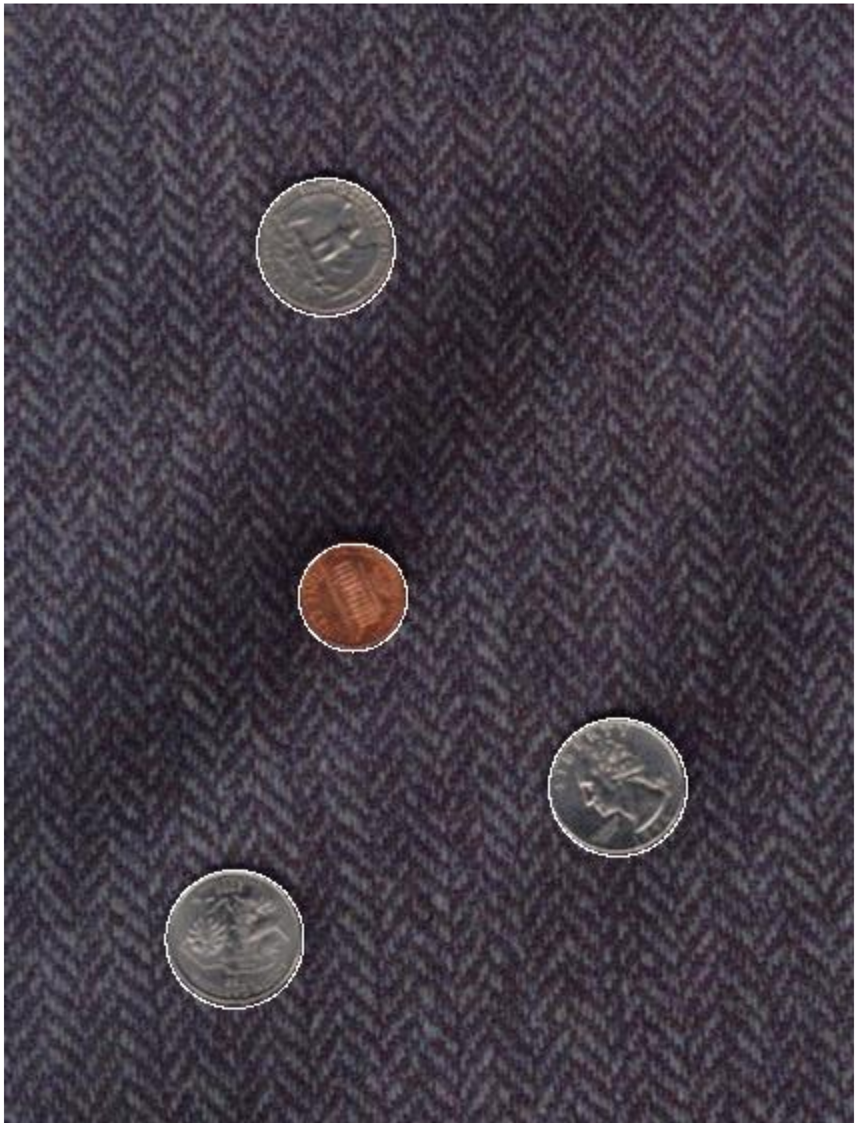


Finding Coins (Continued)

Penn Quarters



Finding Coins (Continued)



Note that because the quarters and penny are different sizes, a different Hough transform (with separate accumulators) was used for each circle size.

Coin finding sample images from: Vivek Kwatra

Hough Transform: Applications

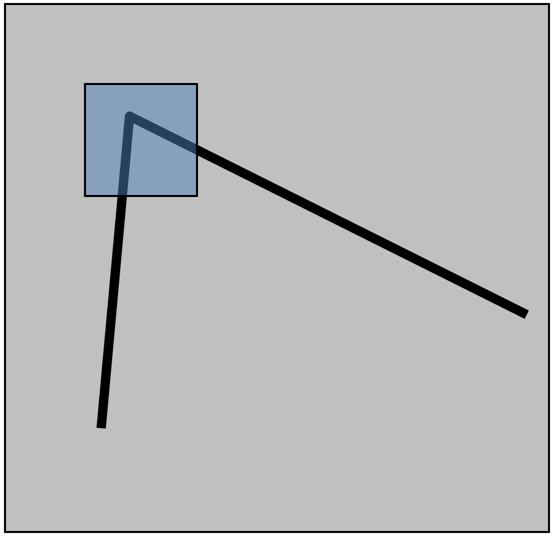
- ❖ Works on Disconnected Edges
- ❖ Effective for simple shapes (lines, circles, Ellipse etc.)
- ❖ Image Recognition
- ❖ Medical field etc.

Corner Detection

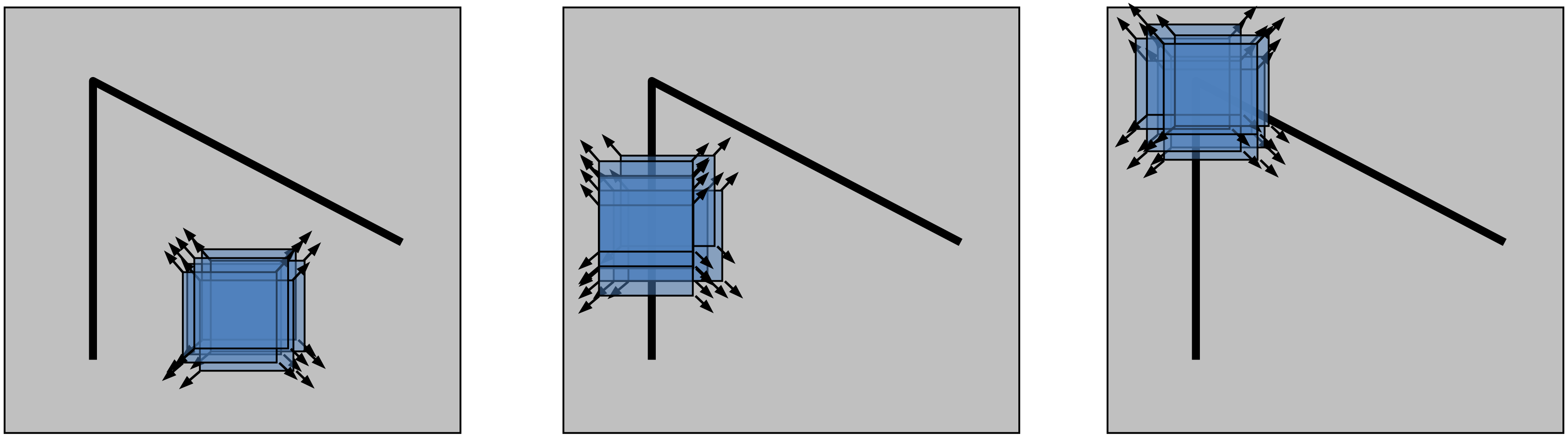
- **A Corner** can be interpreted as **the junction of two edges**, where an edge is a sudden change in image brightness
- A corner is an important feature of an image, which indicates some important points of the image known as **Landmarks Points.**
- It is formed at the boundary when **two bright regions meet**, where the boundary curvature is very high.
- A corner point exhibits the characteristic of having large variations in all directions, unlike edges, where the variation is in only one direction.
- Corner points are also called interest points because the **human visual system quickly identifies landmark points, such as corners, high-contrast regions, and edges.**
- Corners are used in many imaging applications such as **tracking, measurement systems, aligning and stereo imaging.**

Harris Corner Detector

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Harris Detector: Mathematics

Change of intensity for the shift [u,v]:

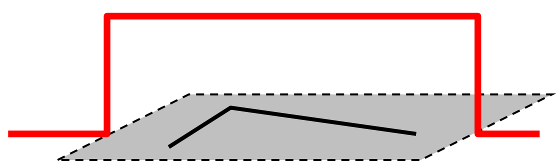
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

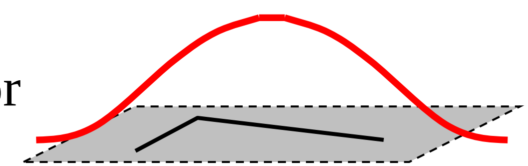
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector

- The autocorrelation function c captures the intensity structure of the neighborhood.
- In addition, it is smoothed by the linear Gaussian-smoothed matrix ***R*** is represented as

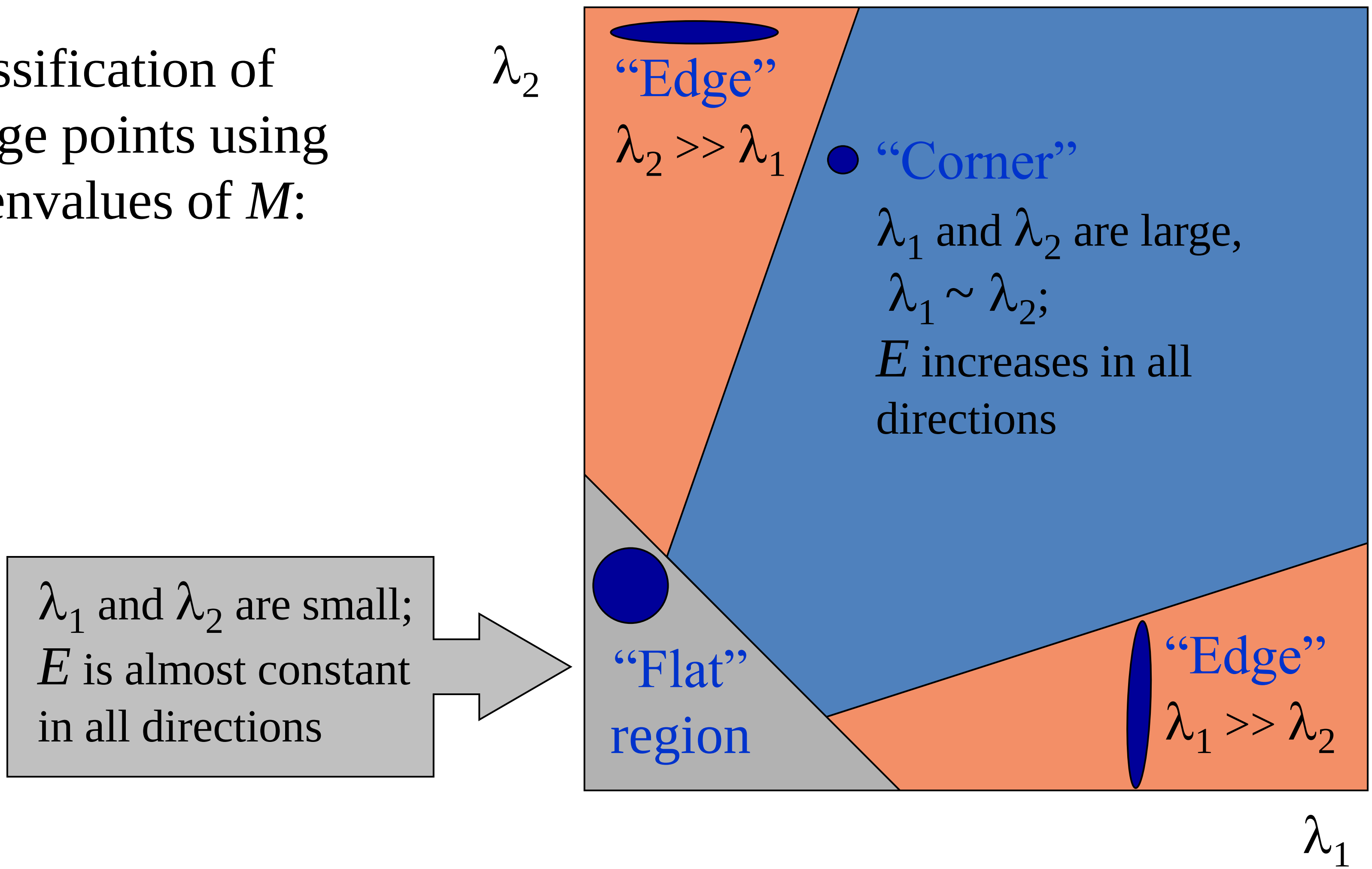
$$R = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

The eigen values of this matrix ***R*** characterize edge strength, and the eigen vectors represent the edge orientation. Assume λ_1 & λ_2 be the eigen values, then

1. If the values of λ_1 & λ_2 are too small, it implies that the window region is flat.
2. If λ_1 is high and λ_2 is small or vice versa, it is an edge.
3. If both λ_1 & λ_2 are high, then it indicates the presence of a corner.

Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector

- In general, the corner strength can be characterized as a corner strength function and represented as A score, Q(u,v), is calculated for each window:

$$Q(u,v)=\det(\boldsymbol{R})-k \times (\mathrm{trace}\ (\boldsymbol{R}^2))$$

The computation of Q(u,v) is easy as the determinant and the trace of the matrix can be expressed in the form of the eigen values are

$$\mathrm{Det}(\boldsymbol{R}) = \prod_{i=1}^{n} \lambda i \quad \text{and} \quad \mathrm{trace}(\boldsymbol{R}) = \sum_{i=1}^{n} \lambda i$$

Harris Corner Detector

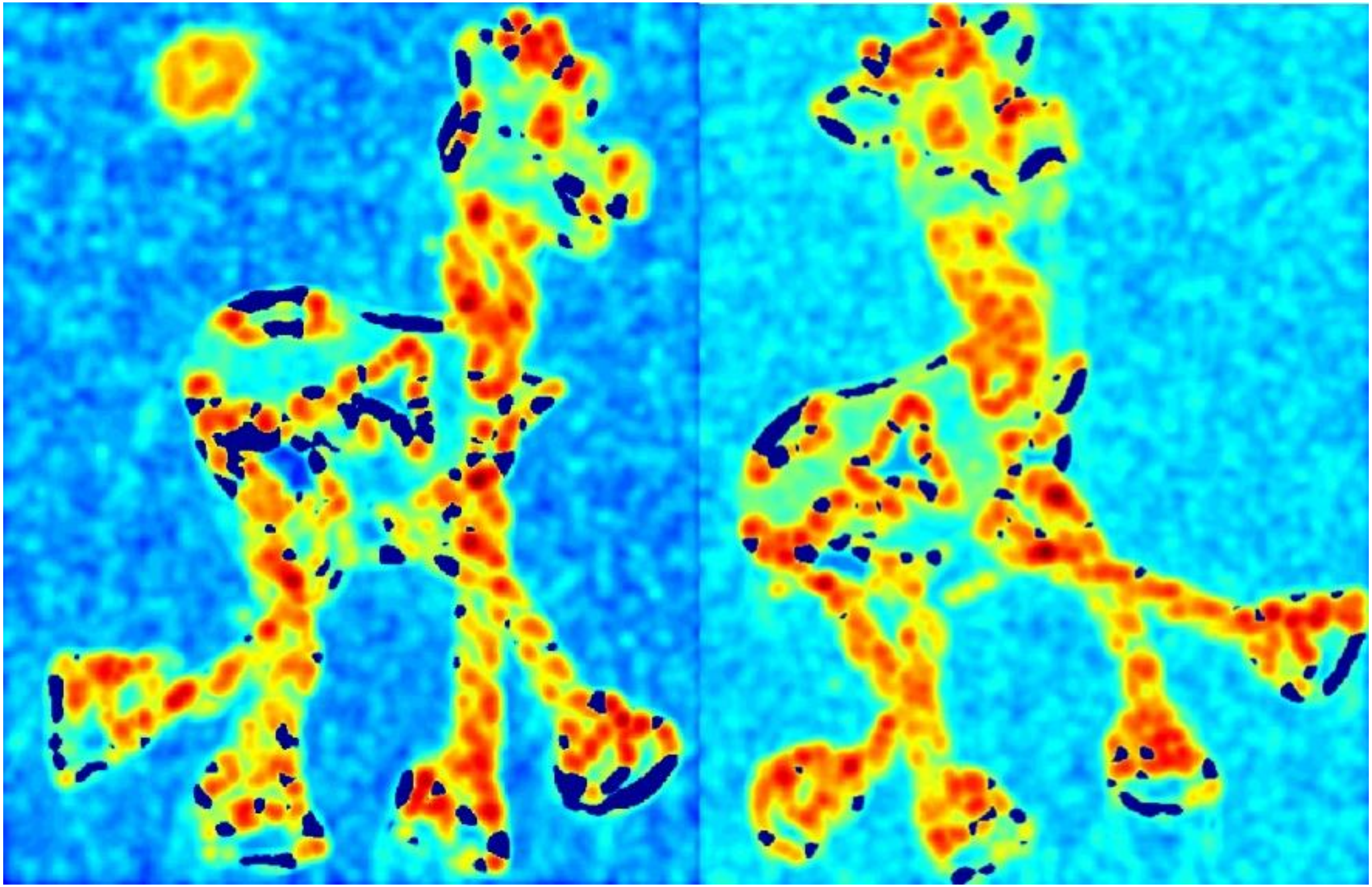
❖ **The Algorithm**:

1. Select an image location (x,y)
2. Identify it is a corner point when strength Q(u,v) > threshold value. Define a window and apply autocorrelation function.
3. Insert into a list of corner points sorted by the corner strength
4. Eliminate the false corners when a weaker corner point lies near a stronger corner.
5. Display all the corner points
6. Exit.

Harris Detector: Workflow

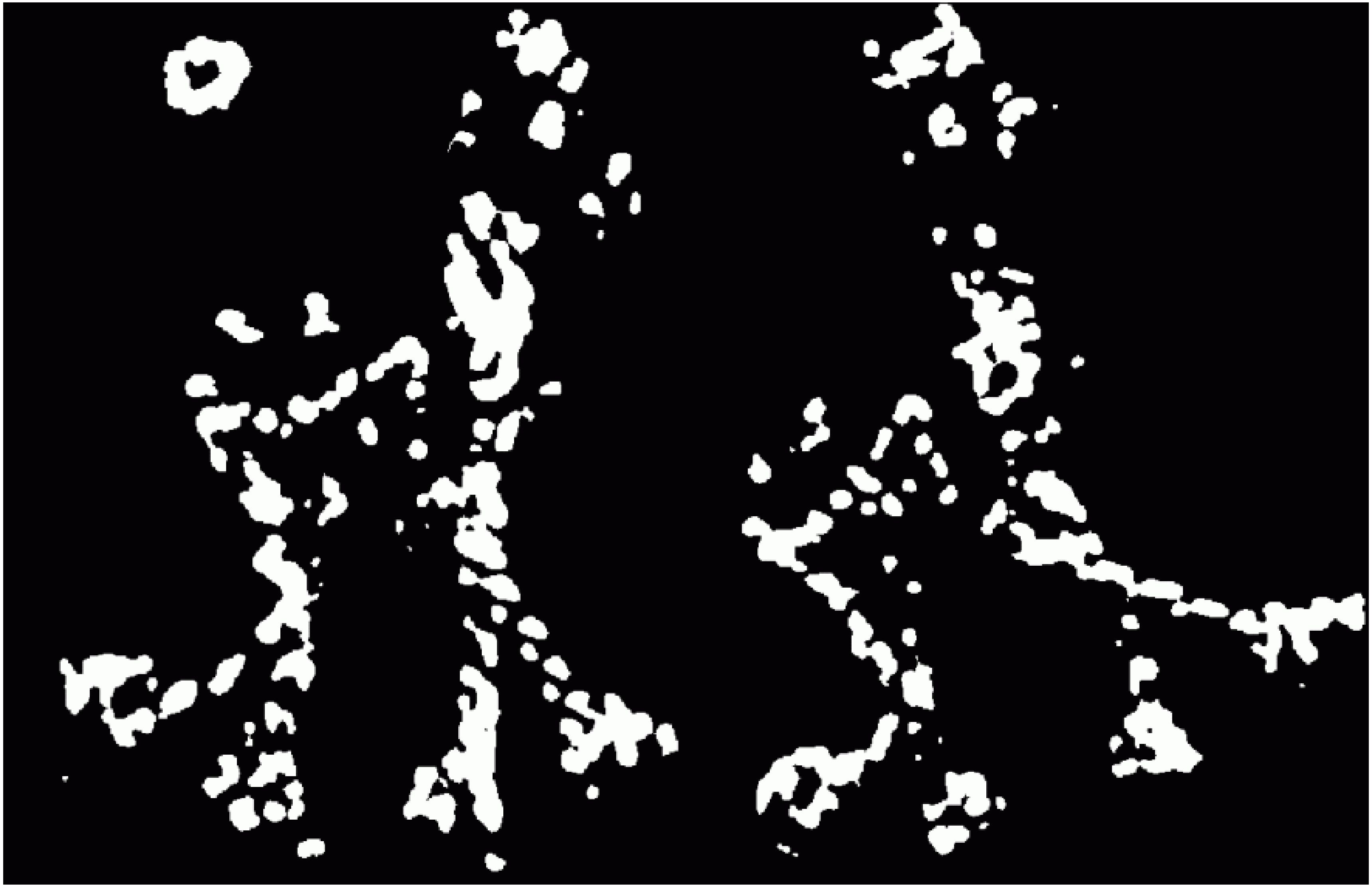
Harris Detector: Workflow

Compute corner response R



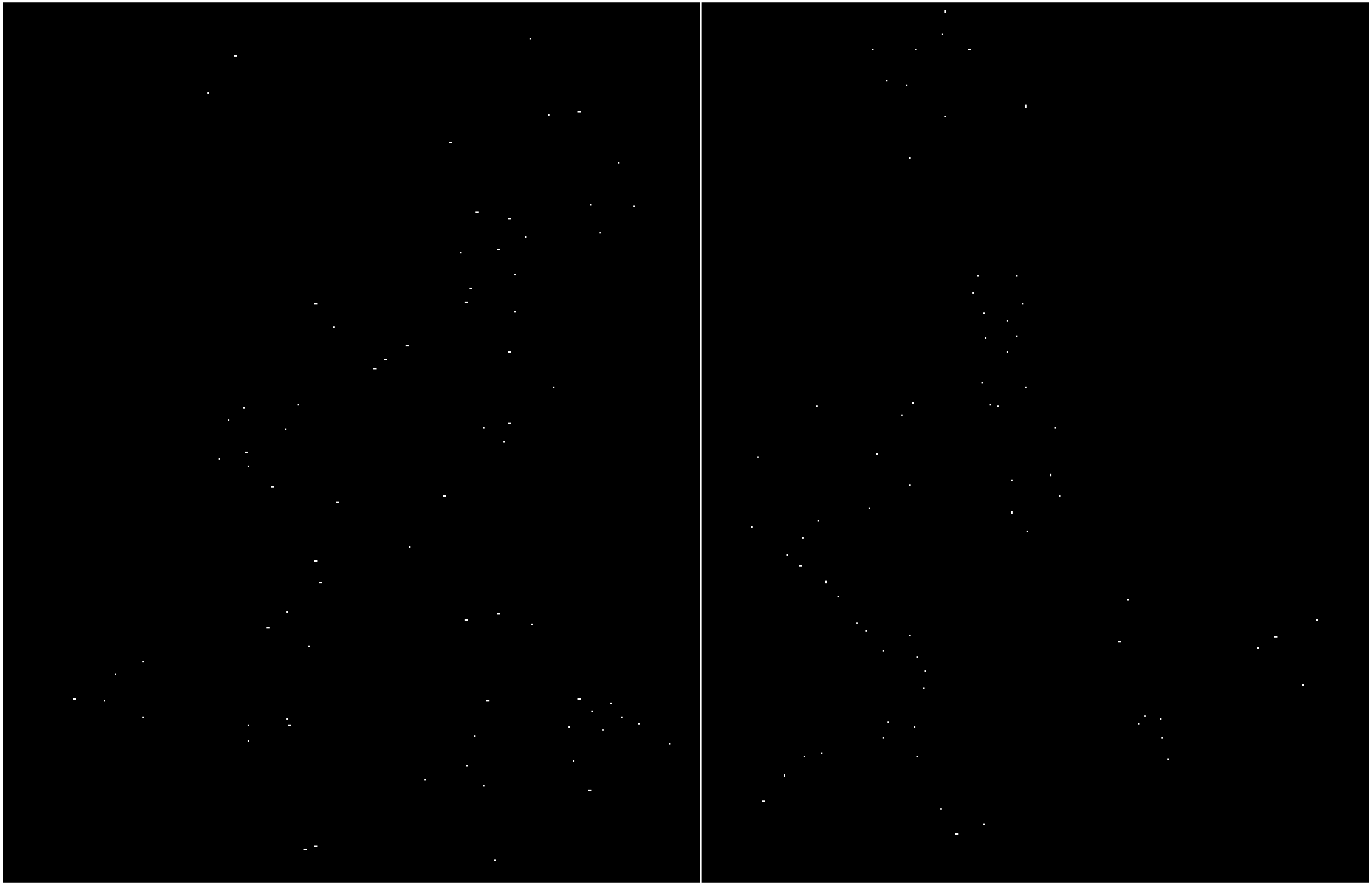
Harris Detector: Workflow

Find points with large corner response: $R >$threshold



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow


Diolch
Kiitos
Sheun
umesc
Kasih
Todah
Shnorhakalutiun
Mamnoon
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ačí
Gamsahapnida
Teşekkür
Dekuju/Dekujeme
Hvala
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
Dhanyavaad
Grazie
Faleminderit
Dakujem
Waad
Daw
Dhanyavadaalu
Kop
Salamat
Merci
Dankie
Go
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Dhonnobaad
ederim
Shukriya
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Euxaristo
Kun
Hain
Dhan
Fyrir
Terima
Enkosi
Danke
dank
daa

Object Representation

By,
Prof (Dr.) P P Ghadekar

Outline

- ❖ Basic Concepts of Object Representation
- ❖ Feature Classifications
- ❖ Chain Code
- ❖ Differential Chain Code
- ❖ Polygonal Approximations
- ❖ Signatures etc

Introduction

- **What is a feature?**
- The first major decision in **object recognition** is **to decide the set of features** that is helpful in Object classification.
- **Any Characteristic or attribute** of an object that **helps to distinguish or discriminate an object from other objects** is called an Image feature.
- Any feature is a function of one or more quantified measurements.
- The object can be characterized by one of the following
- **Natural Features-**Visual Appearances-Brightness, Contrast, texture etc.
- **Artificial Features-**Derived features that are obtained using Image manipulations-Histogram, Frequency spectrum etc.

Introduction

- Feature information of an image is extracted from its content.
- It may include **format descriptors, intrinsic features, spatial features,** relationships among image entities, etc.
- This information is further used for retrieval purpose of images.
- The requirement of feature information is mainly the function of its application domain.
- **For example:** The Medical domain in particular, relative spatial (geometry) relationships among internal image entities contribute as a major factor for image similarity.

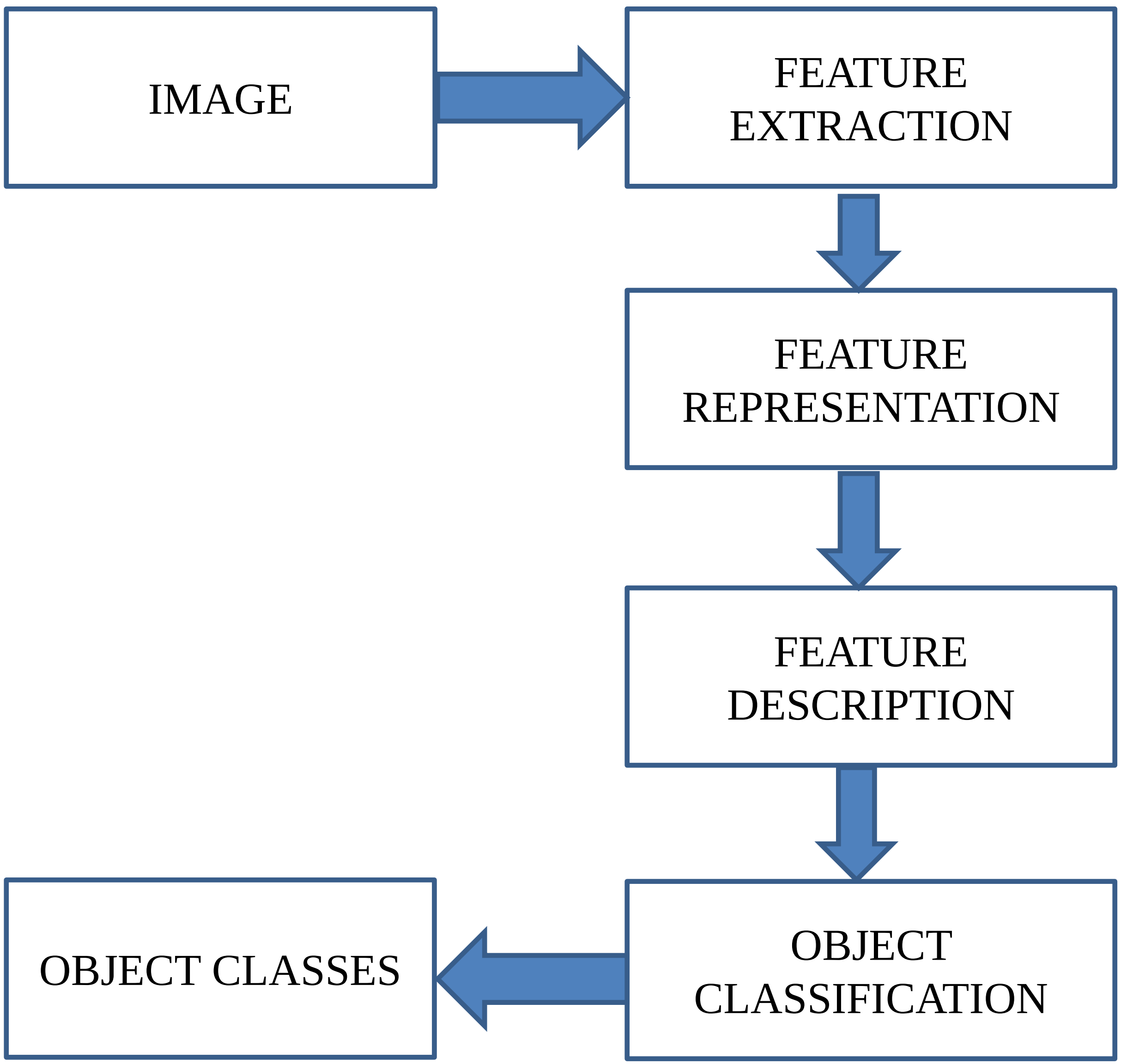
Feature Extraction

- **Feature Extraction** is a process of **transforming the input data** (without redundancy or duplicate value) **into set of features.**
- Instead of full-size input, the features will be extracted from the reduced data containing the relevant information.
- It involves using algorithms to detect and isolate various desired portions or shapes (features) of a digitized image or video stream.
- A feature vector is a vector that contains n measured values as

$$X=[x_1, x_2, \ldots\ldots\ldots\ldots x_n]^T$$

- The classifier takes the feature vector as input and perform the classification.

Image to Object Classification



IMAGE

TO

OBJECT

CLASSIFICATION

Characteristics of a Good Feature

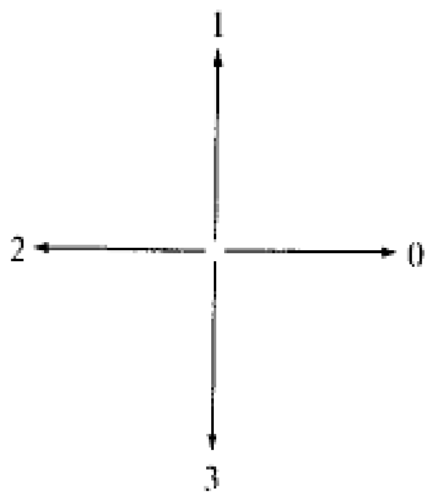
- Robustness
- Shift Invariance
- Rotation Invariance
- Size Invariance
- Mirror, Shear, and Affine Invariance
- Occlusion Invariance
- Discrimination
- Reliability
- Independence
- Resistance to Noise
- Compactness

Boundary Representation

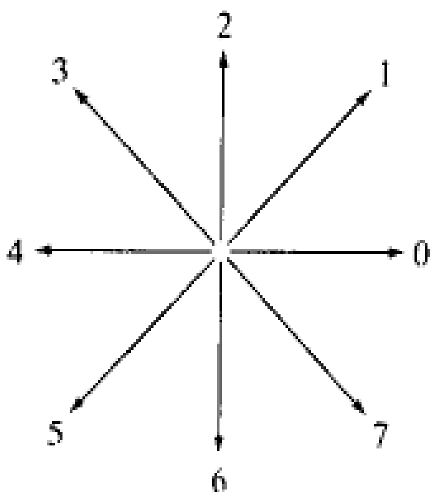
- ❖ After the Segmentation process is over, the segmented points are present in an isolated manner.
- ❖ If the isolated points are connected, it forms a **Boundary**.
- ❖ First, the boundaries should be represented in a suitable form using a data structure so that the analysis can be done.
- ❖ This stage is called **Object Representation**.
- ❖ **Boundary Representation Schemes**
  - Chain Code
  - Differential Chain Code
  - Polygonal Approximation
  - Signatures
  - Bending energy

Chain code

- They are used to represent Boundary by connecting lines having a Specific length & Orientation.



4-DIRECTIONAL CODE



8-DIRECTIONAL CODE

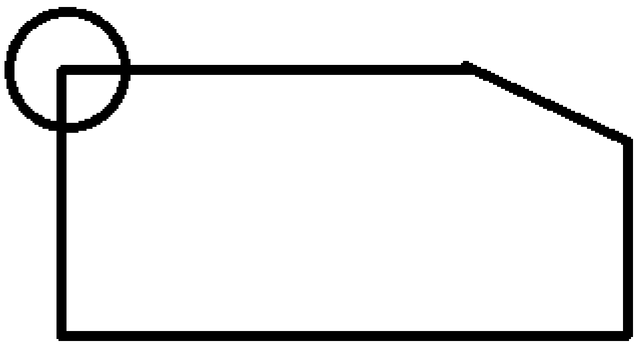
Algorithm for Chain Code

- From left to right start looking for object pixels.
- Start with object pixel 1, naming it as P_0.
- Set d, where d is direction variable.
- Label the pixel as current pixel & previous is labeled as previous pixel.
- Next pixel is obtained by searching the neighborhood in anticlockwise direction.

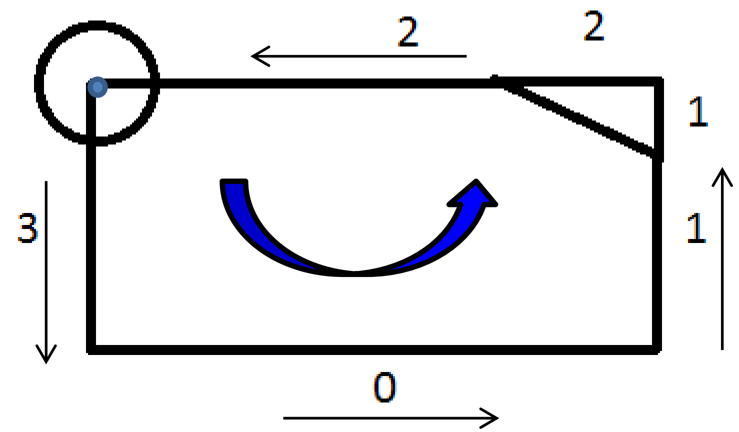
Algorithm for Chain Code

- Search neighborhood
- Now, label the current pixel as previous pixel & next pixel as current pixel.
- Update d.
- If the current boundary pixel P_0 is equal to $P_{n-1,}$ it means , if current pixel = previous pixel then stop. Else go back & repeat the again until starting pixel is reached.
- Label the region.
- Display $P_0 \ldots\ldots . P_{n-1}$ and exit.

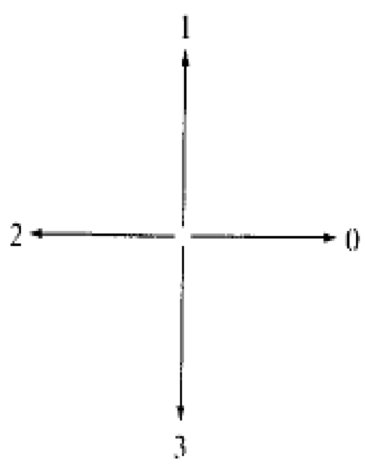
Chain Code



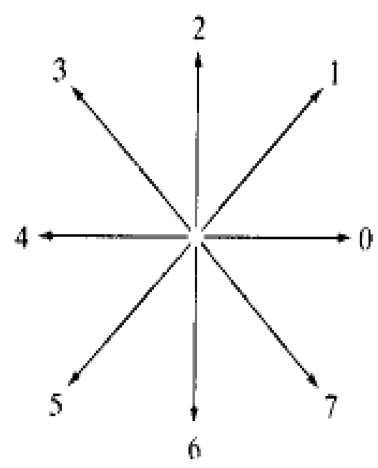
ORIGINAL IMAGE



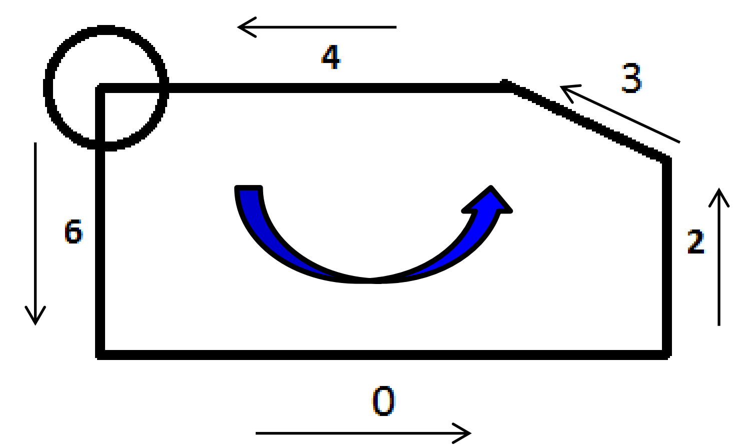
4-DIRECTIONAL CODE 301122



4-DIRECTIONAL CODE



8-DIRECTIONAL CODE



8-DIRECTIONAL CODE 60234

Advantage of Chain Code

- Information of interest is preserved.
- It is a compact representation of an image.
- A chain code is prior condition for image analysis.
- It is translation invariant.
- It represents any change occurring in direction, indicating the corner points.
- **Drawback**-It is not rotation & scale invariant.

Problems in Chain Code

1. **Starting point of code determines Chain code.**

❖ So, any change in Rotation or Size affects the Chain Code in unnormalized form.

2. **Chain code is Noise Sensitive.**

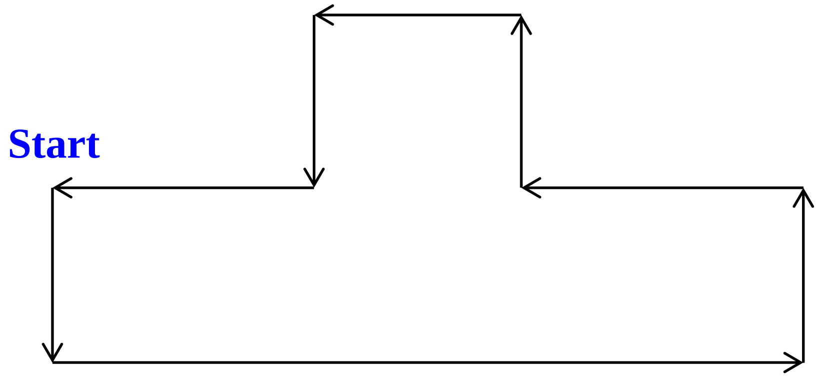
❖ To solve this problem , resampling of boundary is done to smooth out small variation.

Differential Chain Code

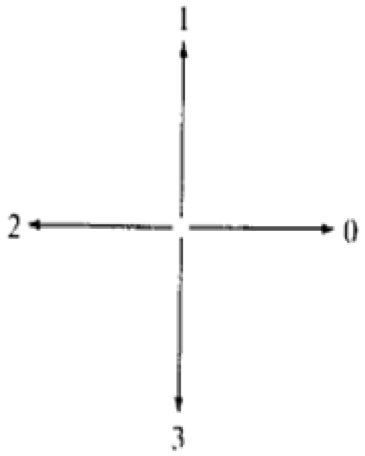
- **Normalization solves the problem of the starting point.**
- The way to normalize the chain code is to take the first difference of the chain code.
- **What is the first difference?** –The first difference of the chain code is obtained by taking two numbers of the chain code and calculating the number of transitions required to reach the second number from the first number in the counter clockwise direction.
- **Step-1:** Consider the code 3 0 1 1 2 2. In 4-directional chain code, it takes one transition in the counter clockwise direction to start from 3 and to reach 0.
- Therefore for the chain code-3, 0 0 1 1 2 2, the first difference would be 3-0, 0-1, 1-1, 1-2, 2-2, that is 1 1 0 1 0
- **Step-2:**This code can be treated in a circular fashion and the normalized code can be obtained.
- How? The transition from the last code to the first code in the counter-clockwise direction is obtained and this is the first code.
- Ex- Transition from 2 & 3 is 1. Therefore the final chain code is **1 1 1 0 1 0**

Differential Chain Code

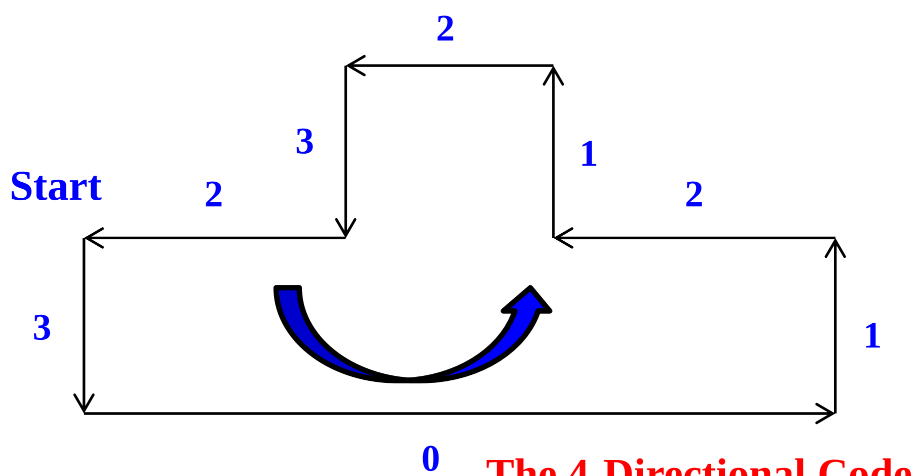
- Consider the following shape and write the corresponding 4-directional and 8-directional codes. Perform Circular Normalization.



- Solution-Applying the 4-Directional code, We obtain



4-DIRECTIONAL CODE



The 4-Directional Code is 3 0 1 2 1 2 3 2

Differential Chain Code

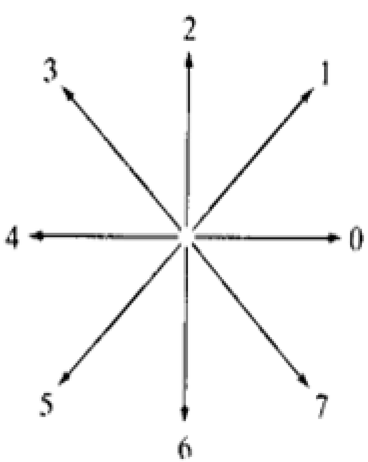
- The 4-directional code is 3 0 1 2 1 2 3 2
- The normalize circular code, using this example, involves the transitions

2-3, 3-0, 0-1, 1-2, 2-1, 1-2, 2-3, 3-2

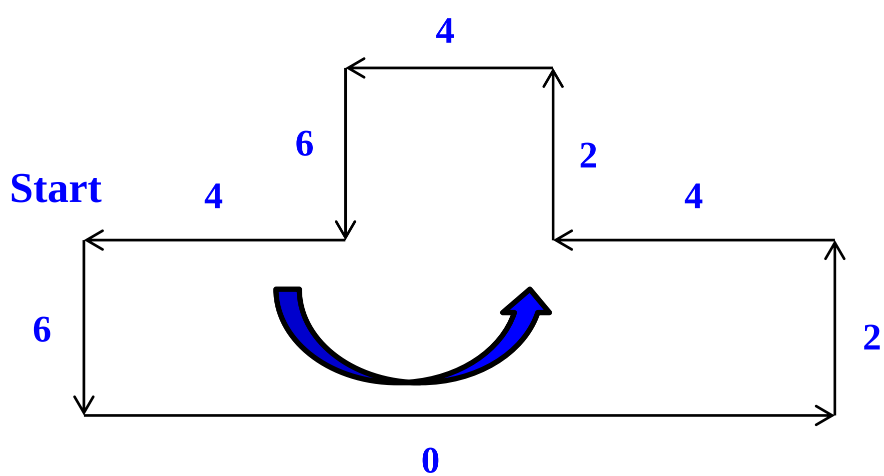
The first transition 2-3 is between the last and the first elements of the chain code. The other transitions are between an elements and its subsequent elements. This results in the following normalized code:

1, 1, 1, 1, 3, 1, 1, 3=1 1 1 1 3 1 1 3

Applying the 8-directional code, we obtain



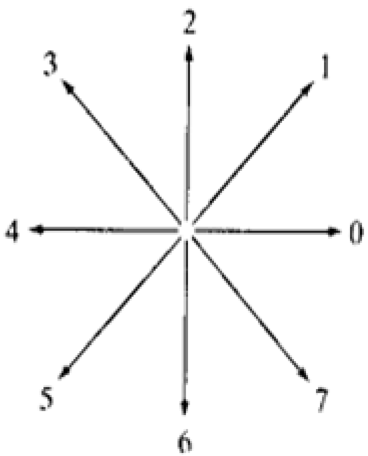
8-DIRECTIONAL CODE

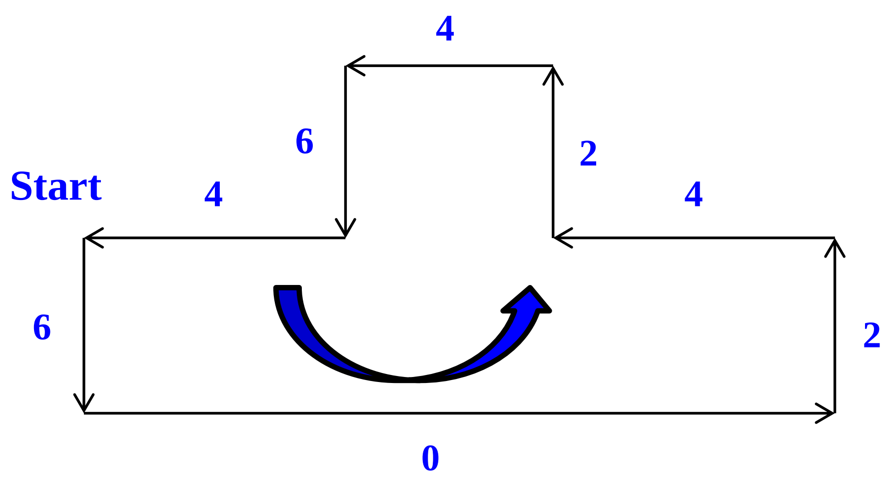


The 8-directional code is 6 0 2 4 2 4 6 4

Differential Chain Code

Appling the 8-directional code, we obtain





8-DIRECTIONAL CODE

The 8-directional code is **6 0 2 4 2 4 6 4**

The normalization process is similar to 4-directional coding

The Normalization Circular code is **4-6, 6-0, 0-2, 2-4, 4-2, 2-4, 4-6, 6-4**

= 2, 2, 2, 2, 6, 2, 2, 6 = 2 2 2 2 6 2 2 6

Polygon Approximation

- It is a process of connecting points of interest with approximation. Using fewest possible points this method is applied.

- Due to approximation loss of information occurs.

- The main reason to use this method is to approximately reduce the code for polygon.

Polygon Approximation

❖ There are 2 types of approximation:

1. *Fit & split*

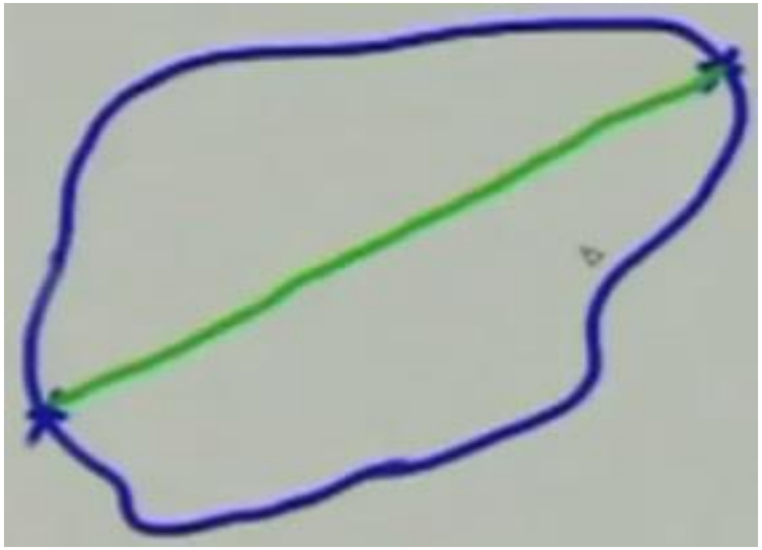
2. *Merging*

Polygon Approximation

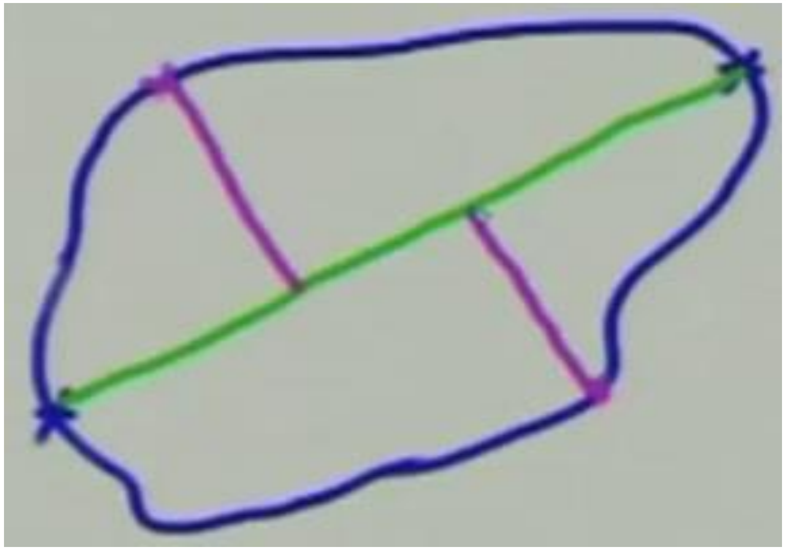
- The simplest way of approximation is called the Fit and split method.
- **Procedure**

1. Divide the boundary into small segments and fit a line for each segment.
2. Draw a line from any points to its large distance point. Identify the longest line segment that connects the two large distance points.
3. Compute the perpendicular distance from the boundary to the line segment.
4. Check the perpendicular distance with a threshold. If the threshold is crossed, then the line is split.
5. Identify the vertices that are very close to the prominent inflection points of the curve. An inflection point is a point on a curve where curvature changes its sign.
6. Connect the points to get the approximate polygons.
7. Repeat the process for the new line until there is no longer a need for a spilt.

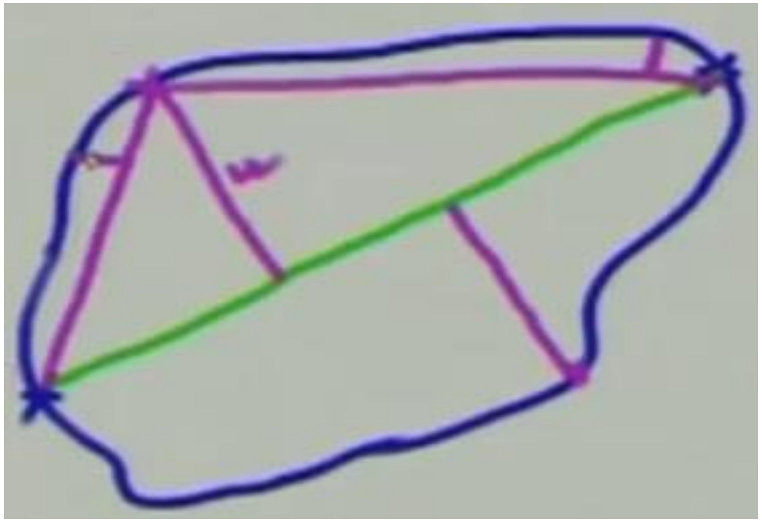
Fit & Split method



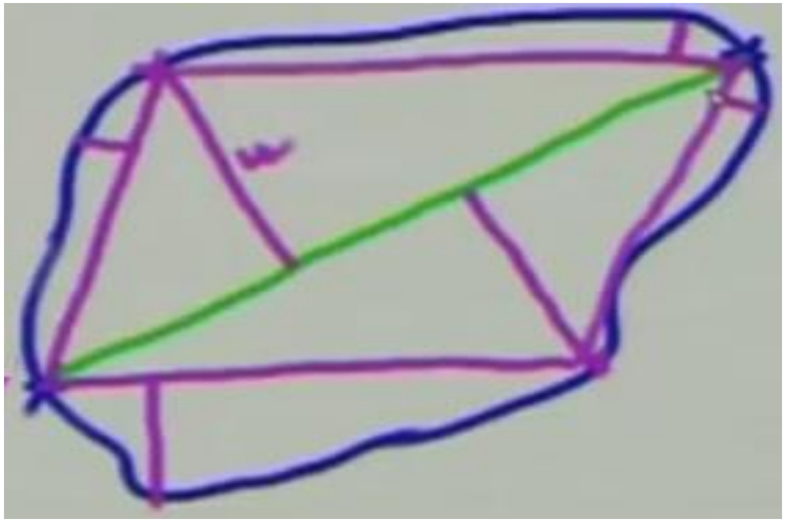
(A)



(B)

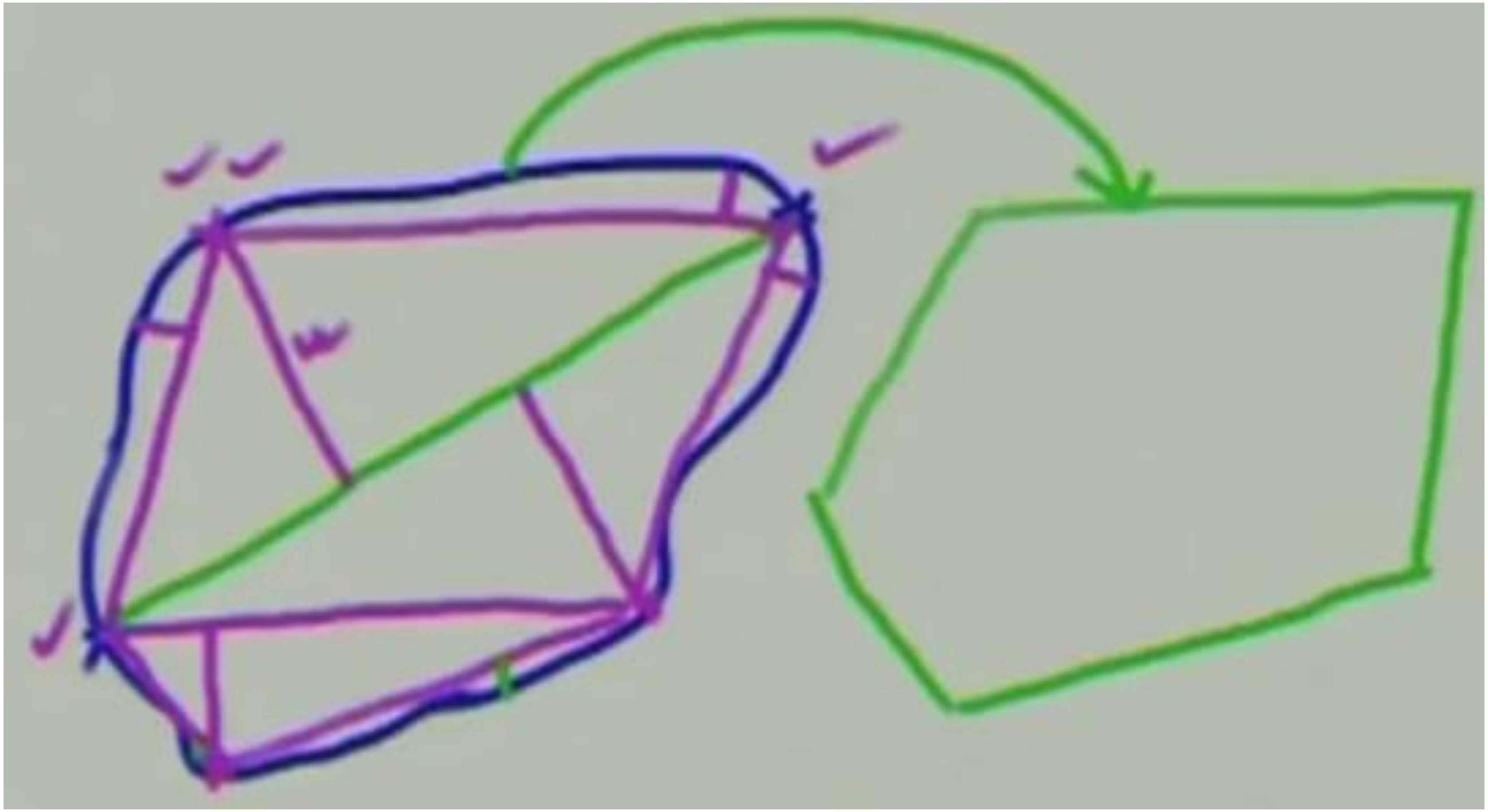


(C)



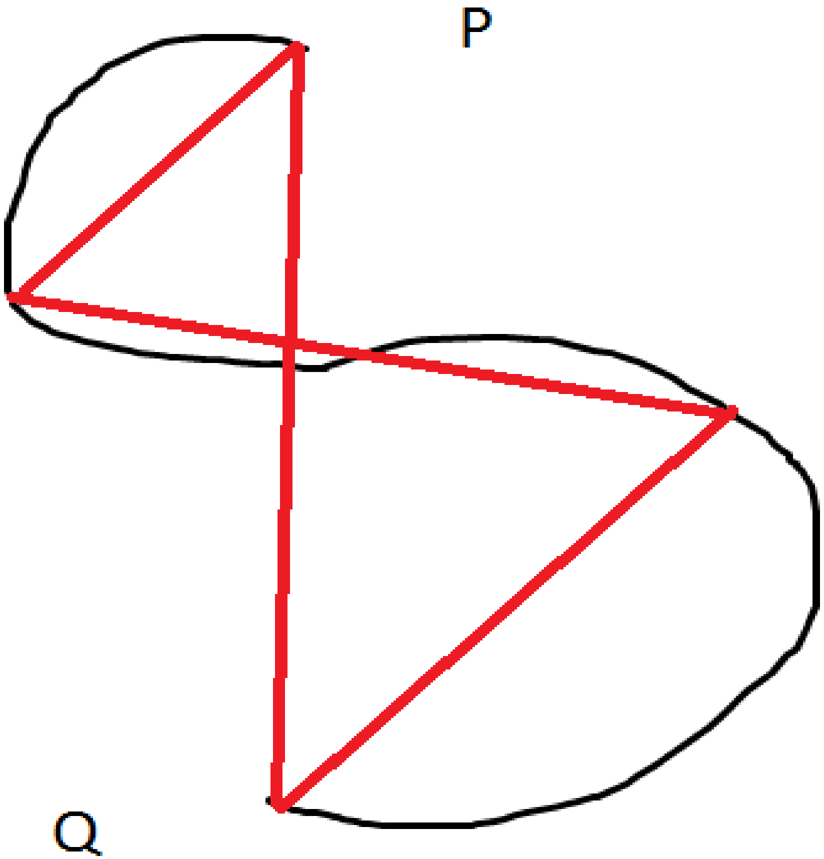
(D)

Fit & Split method



Final Output

Fit & Split Process for S Shape Image



Fit & Split method

❖ **This method is useful for following reasons:**

i. Noise elimination

ii. Absence of relevant feature

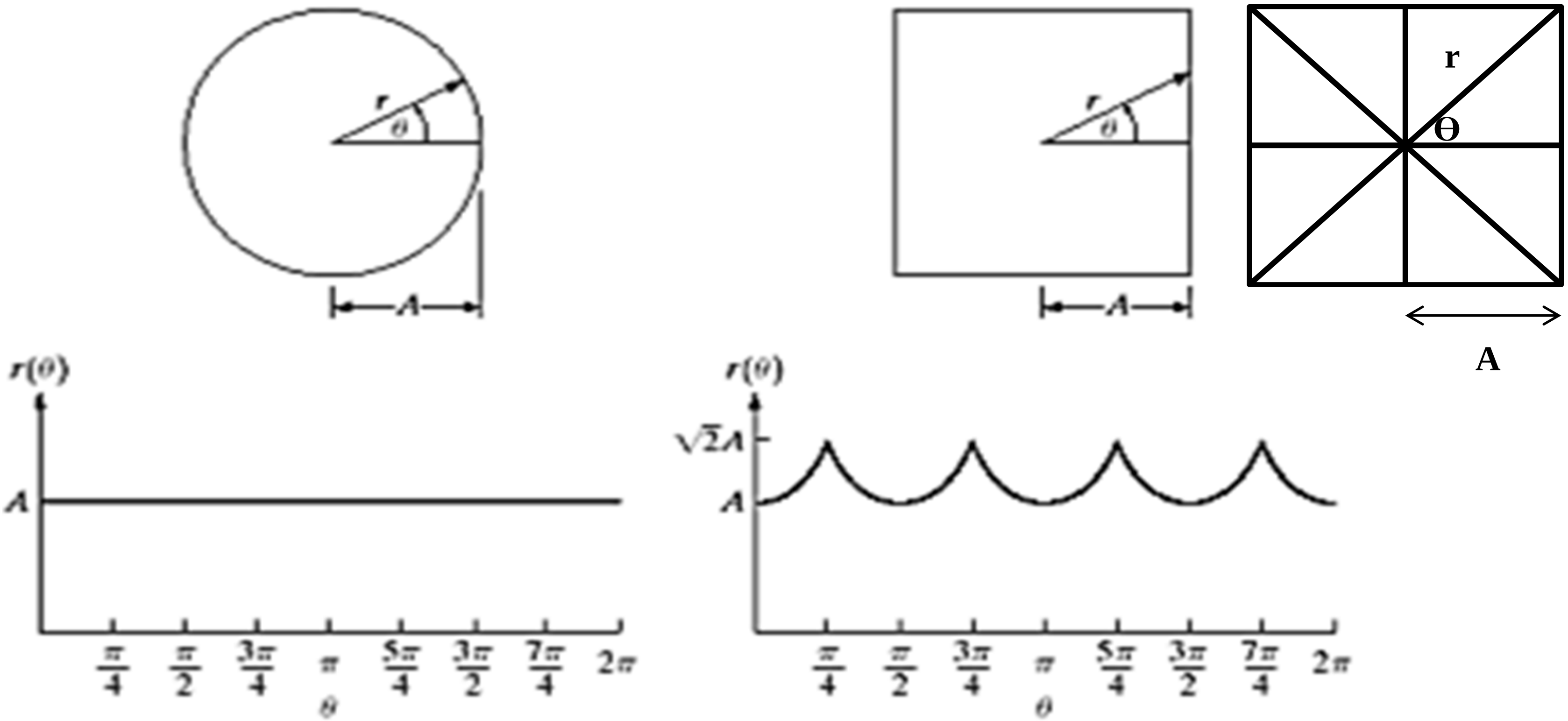
iii. Absence of change in vertices

iv. Simplification of shape

Signatures

- Sometimes its required **to represent 2D functions in 1D for fast & easy object matching.** This is done using **Signature Function.**
- It is **invariant to translation** but not **invariant of rotation and scaling.**
- It is a **plot of the distance from the centroid to the boundary as a function of the angle.**
- The starting point can be chosen as a point in the Eigen axis that is at the greatest distance from the centroid.
- The other way to choose this point is by plotting the **angle between the tangent line & the reference line**
- See example on next slide.
- **Centroid**-the centroid or geometric center of a plane figure is the arithmetic mean position of all the points in the figure.
- **Tangent line**-A line that just touches a curve at a point, matching the curves slope there.

Signatures-Other Example



It is a plot of the distance from the centroid to the boundary as a function of the angle.

A-Radius of the circle, r is the distance between the centroid to object boundary, Θ is the angle with respect to reference axis. Θ along the x-axis and r along y axis. Distance Vs angle.

Bending Energy

- Its another form of signature where $\varphi - s\ curve$ is used. $\varphi - s\ curve$ is a plot of tangent φ as a function of length around a curve.

- So this method store the value of tangent at each point of curve.

- It is useful because it is rotation invariant, translation invariant & in normalized form its scale invariant.

Bending Energy

- The integration of all squared curvature value along the entire contour gives a single Descriptor called **Bending Energy.**
- This is captured in the discrete domain as the sum of squares of the border curvature c(x) over the boundary of length L. It can be described in the discrete domain as

Equation:

$$\text{bending energy} = \frac{1}{L}\sum_{i=1}^{L} c^2(x)$$

Main problem with this technique is it does not allow reconstruction of the original object

Bending Energy

The Curvature can also be obtained from Chain Code. The Chain code and the slope are given as follows

Chain code	0	1	2	3	4	5	6	7
θ	**0**	**45^0**	**90^0**	**135^0**	**180^0**	**-135^0**	**-90^0**	**-45^0**
tan θ	0	1	$\pm\infty$	-1	0	1	$\pm\infty$	-1

The Curvature is given as $\Delta\theta = \theta_{i+1} - \theta$


Diolch
Kiitos
Sheun
umesc
Dekuju/Dekujeme
Kasih
Todah
Mamnoon
Shnorhakalutiun
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ači
Gamsahapnida
Te ekkür
Hvala
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
or
al
Dhanyavaad
Dakujem
Waad
Daw
Dhanyavaadalu
Kop
Salamat
Merci
Dankie
Go
Grazie
Faleminderit
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun
Hain
Dhan
daa

Image Statistical properties

By

Prof (Dr.) P P Ghadekar

OUTLINE

- **On the Completion Student will be able to understand**
- Brightness
- Contrast
- Mean
- Standard deviation
- Probability distribution function
- Implementing statistical properties using MATLAB.
- Implementing statistical properties using ‘C’.
- Verification of MATLAB results with C.

STATISTICAL PROPERTIES

- Statistical properties provide useful information about an image at a glance.
- For example “mean” tells us the average value of all the pixels hence we know that the image is a dark one or light one.
- They helps us to search information in numerical form.
- Variance gives an idea of the distribution of the various intensities about the mean value.

STATISTICAL PROPERTIES

- **Brightness:**
- It is the perceived darkness or brightness and is defined as the average of all the pixels within an image.
- The brightness indicates the mean value of the image.

Statistical properties

- **Contrast:**
- It gives the range of gray level variation in an image. Zero gray level indicates dark intensity.
- It is the average variation in gray level within an image.
- Contrast value i.e. C varies between 0 to1.
- C=1 means high contrast image, i.e maximum and minimum gray levels are present in the image.
- C=0 indicates only one gray level is present in the image.
- Contrast indicates Standard Deviation.

NUMERICAL DESCRIPTIONS

Let y denote a quantitative variable, with observations y_1 , y_2 , y_3 , … , y_n

a. Describing the *center*

Median: Middle measurement of ordered sample

Mean:

$$\bar{y} = \frac{y_1 + y_2 + \ldots + y_n}{n} = \frac{\Sigma y_i}{n}$$

Example: Annual per person carbon dioxide emissions (metric tons) for n = 8 largest nations in population size

Bangladesh 0.3, Brazil 1.8, China 2.3, India 1.2, Indonesia 1.4, Pakistan 0.7, Russia 9.9, U.S. 20.1

Ordered sample:

Median =

Mean $\bar{y}$ =

Mean-Image

- **Mean**
- It is the measure of average gray level or brightness in an image.
- Let r denote a discrete random variable representing discrete gray level in the range [0, L – 1] and $P(r_i)$ denote the normalized histogram component corresponding to the i^{th} value of r. $P(r_i)$ can be viewed as an estimate of the probability of occurrence of gray level r_i
- Mean is given by $$M = \sum_{i=0}^{L-1} r_i \cdot P(r_i)$$
- All this formula says is "Add up all the numbers and then divide by how many there are".

Mean

- Unfortunately, the mean doesn't tell us a lot about the data except for a sort of middle point.
- For example, these two data sets have exactly the same mean (10), but are obviously quite different

$$[0\ 8\ 12\ 20]\ and\ [8\ 9\ 11\ 12]$$

- So what is different about these two sets? It is the *spread* of the data that is different.
- The Standard Deviation (SD) of a data set is a measure of how spread out the data is.

Standard Deviation

- **Standard Deviation:**
- How do we calculate it? The English definition of the SD is: **"The average distance from the mean of the data set to a point".**
- The way to calculate it is to compute the squares of the distance from each data point to the mean of the set, add them all up, divide by N-1, and take the positive square root. As a formula-

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

Where S is the usual symbol for standard deviation of a sample.

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Where $\overline{X}$ is the mean of data set X

Standard Deviation

- Standard Deviation is measure of average contrast of an image. So if deviation decreases contrast of an image decreases.
- It measures average peak to peak gray level deviation of noise.
- The Standard deviation is simply defined as square root of variance, where variance is the 2nd moment of r about its mean and is defined as

$$\mu_2(r) = \sum_{i=0}^{L-1} \left(r_i - M\right)^2 \cdot P(r_i) = \sigma^2(r)$$

Therefore, Standard Deviation is given by $\sqrt{\sigma^2(r)}$

Variance

- Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation. The formula is this

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

- Standard Deviation and Variance both these measurements are measures of the spread of the data.
- Standard deviation is the most common measure, but variance is also used.

WHY N-1 IS USED IN STANDARD DEVIATION

- In general, if your data set is a sample data set, ie. you have taken a subset of the real-world (like surveying 500 people about the election) then you must use (n-1) because it turns out that this gives you an answer that is closer to the standard deviation that would result if you had used the entire population, than if you'd used .

Application of Mean and Deviation:

- Image Enhancement.
- For adjustment of intensity of an image.
- For local image enhancement, where local mean and variance are used as basis for making changes that depend on image characteristics in predefined region.

COVARIANCE

- Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set *independently* of the other dimensions.
- However, it is useful to have a similar measure to find out how much the dimensions vary from the mean *with respect to each other*.
- Covariance is such a measure. Covariance is always measured between 2 dimensions.

COVARIANCE

- If you calculate the covariance between one dimension and itself, you get the variance.
- So, if you had a 3-dimensional data set (x,y,z), then you could measure the covariance between the (x and y) dimensions, the (y and z) dimensions, and the (x and z) dimensions.
- Measuring the covariance between (x and x) or (y and y) or (z and z) would give you the variance of the x, y and z dimensions respectively.

Covariance

- The formula for covariance is very similar to the formula for variance. The formula for variance could also be written like this

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

- where I have simply expanded the square term to show both parts. So given that knowledge, here is the formula for covariance

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Thanks

Image compression

By,

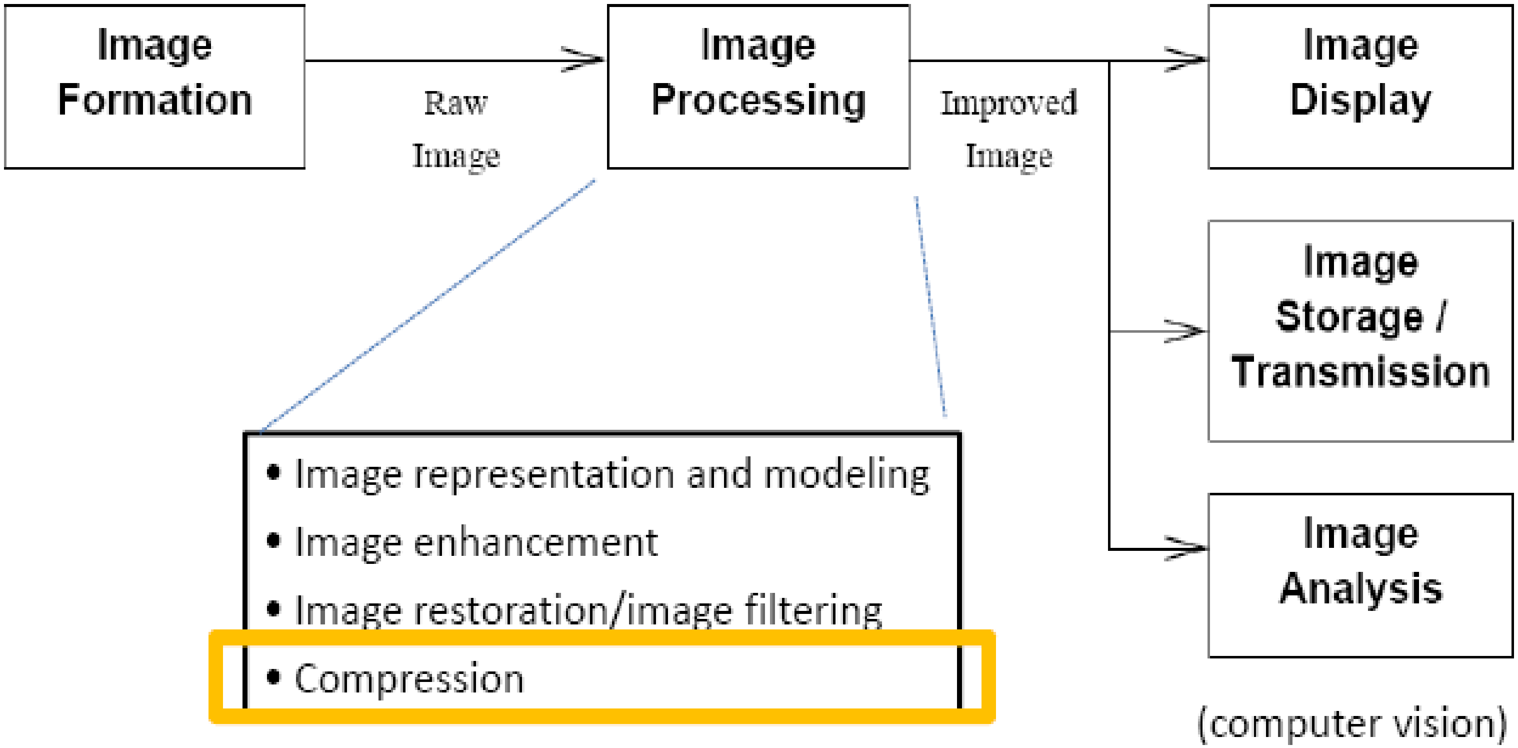
Prof (Dr.) P P Ghadekar

OUTLINE

- On completion of this topic students will be able to understand
- Basics concepts of Image Compression.
- Run Length coding
- Entropy coding
- Huffman coding

WHAT IS IMAGE PROCESSING?

Image processing is the application of 2D signal processing methods to images



DEFINITION OF IMAGE AND VIDEO COMPRESSION

Image and video data compression refers to a process in which the amount of data used to represent image and video is reduced to meet a bit rate requirement (below or at most equal to the maximum available bit rate).

- The quality of the reconstructed image or video satisfies a requirement for a certain application.
- The complexity of computation involved is affordable for the application.

The basis of compression process is removal of redundant data.

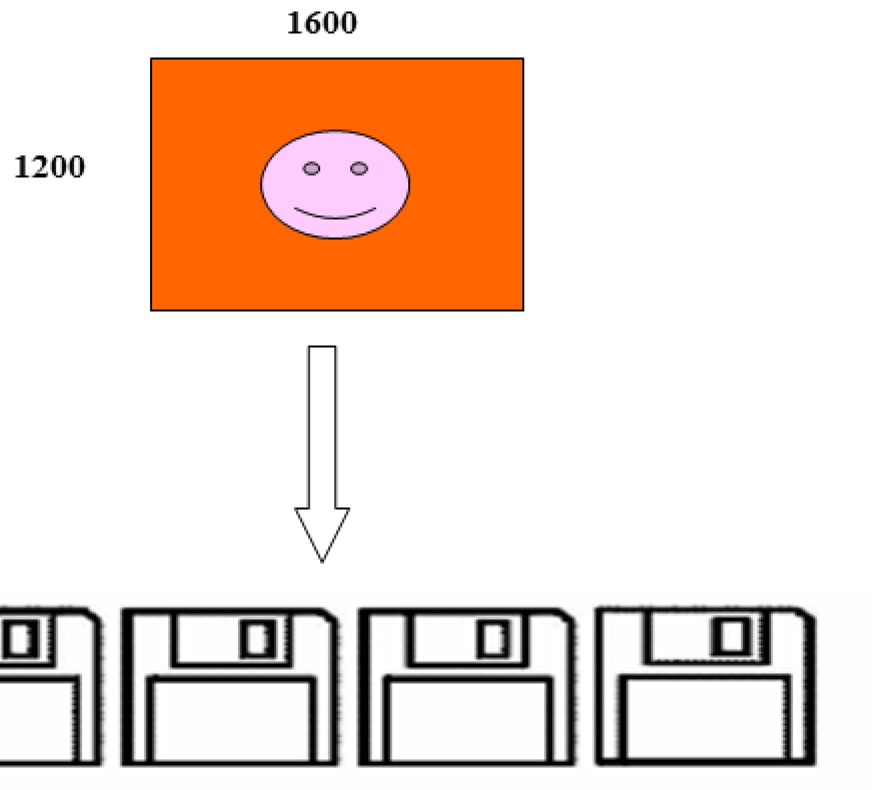
WHAT IS THE NEED FOR COMPRESSION

- In terms of storage, the capacity of a storage can be effectively increased.
- In terms of communications, the bandwidth of a digital communication link can be effectively increased .
- At any given time, the ability of the internet to data is fixed. Thus, if data can effectively be compressed wherever possible, significant improvement of data throughput can be achieved.
- Many files can be combined into one compressed documents making sending easier.

NEED FOR COMPRESSION

- To minimize the Storage Space
- To enable higher rate of Data transfer.
- There is still a need to develop newer and faster algorithms adapted to image data

NEED FOR IMAGE COMPRESSION



FUNCTIONALITY IN VISUAL TRANSMISSION AND STORAGE

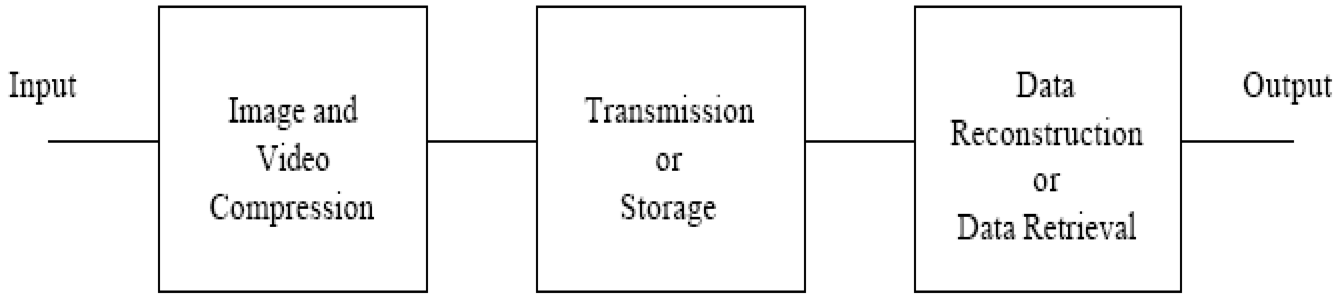


Image and video compression for visual transmission and storage

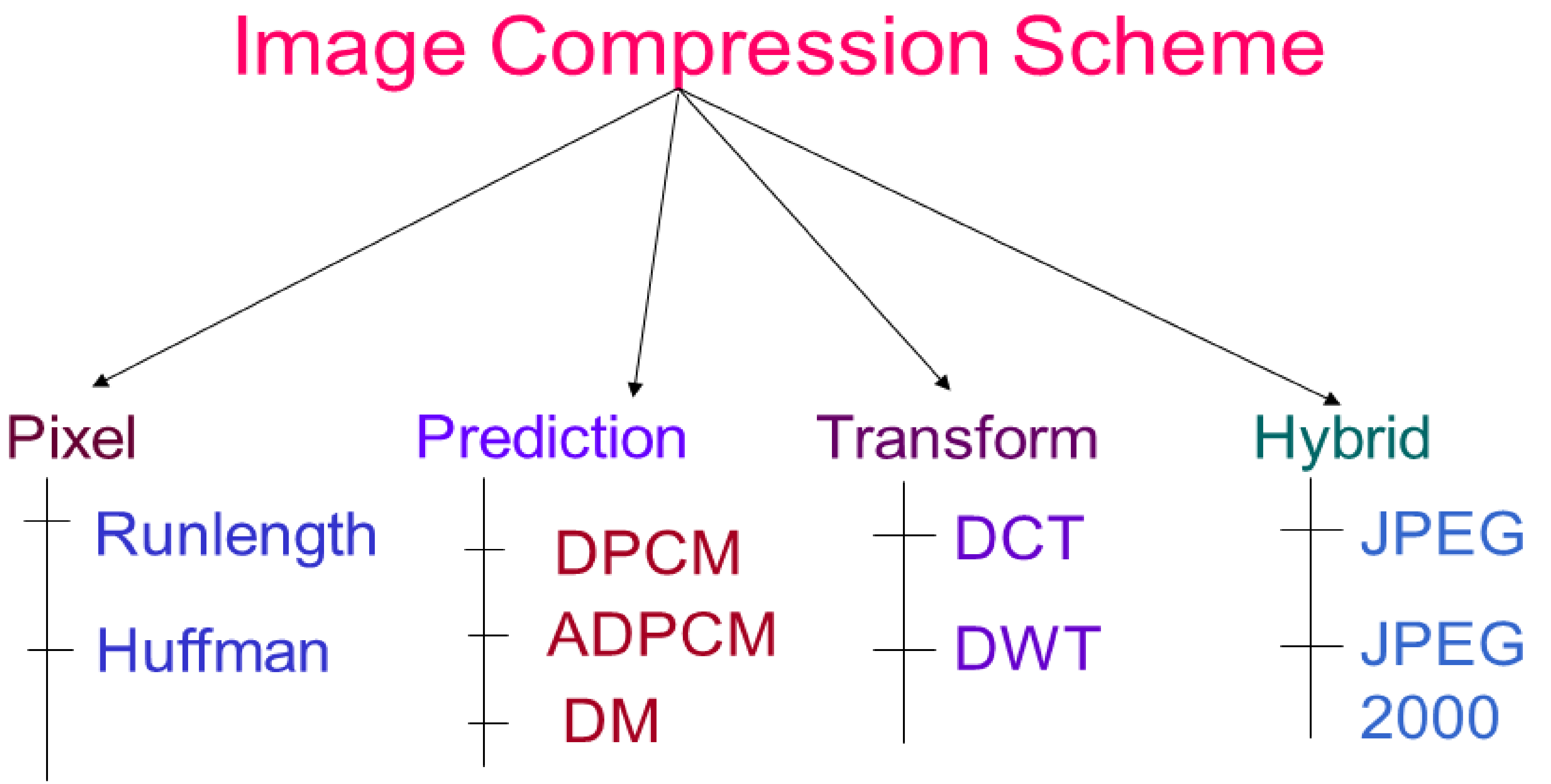
Types of compression

- **There are two types of compression**
- **Lossless compression-** It can recover the exact original data after compression.
  - It is used mainly for compressing database records, spreadsheet or word processing files, medical imaging, where exact replication of the original is essential.
- **Lossy compression-** It will result in a certain loss of accuracy for a substantial increase in compression.
  - Lossy compression is more effective when used to compress graphic images and digitized voice where losses outside visual or aural perception can be tolerated.

HOW TO ACHIEVE COMPRESSION

- Minimizing the Redundancy in the Image.
- Types of Redundancy-Mainly two types
- **Statistical Redundancy-** Interpixel Redundancy & Coding Redundancy.
- Interpixel Redundancy-Spatial Redundancy & Temporal Redundancy.
- **Psychovisual Redundancy.**

Image Compression techniques



TYPES OF COMPRESSION: LOSSLESS

Basic Idea:

- Convert each gray-value (symbol) into a binary sequence (codeword)
- Codewords may have different lengths for different gray-values.

Exploit statistical redundancy

- Assign shorter binary sequences to more common gray levels, and longer codewords to less common ones.
- How do we make this assignment quantitatively? Use the notion of Entropy.

TYPES OF COMPRESSION: LOSSLESS

- Perfectly reproduce the image at the decoder(MSE=0, PSNR=Inf)
- Reduction in bit rate is usually limited.
- Can not guarantee a fixed rate.
- Applications: –Fax Machines–Medical Images

Lossless compression methods

- Run length Encoding (RLE)
- Huffman coding
- Arithmetic coding
- LZ and LWZ coding
- Predictive coding
- Variable length coding etc

Types of Compression: Lossy

- Idea: some deviation of decompressed image from original is often acceptable.
- Human visual system might not perceive loss or tolerate it

  Thus, omit irrelevant details that humans cannot perceive.
- No need to represent more than the visible resolution in:
- Space
- Time
- Brightness
- color

Exploiting Limits of Color Vision

- Human visual system has much lower acuity for color hue and saturation than for brightness
- Use color transform to facilitate exploiting that property

Luminance component

$$\begin{pmatrix} x_Y \\ x_{Cb} \\ x_{Cr} \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.0813 \end{pmatrix} \cdot \begin{pmatrix} x_R \\ x_G \\ x_B \end{pmatrix}$$

Chrominance components

RGB components (γ-predistorted)

Compression: Lossless & Lossy

- Lossy Compression-
  Does not perfectly reproduce the image at the decoder (MSE>0)
- Much higher compression than with lossless compression.
- Can guarantee a fixed rate.
- Lossy compression

  It is used widely for natural images (e.g. JPG) or videos (E.g MPEG)

RUN LENGTH CODING

- Run-length coding is a very widely used and simple compression technique which does not assume a memoryless source
  - We replace runs of symbols (possibly of length one) with pairs of (*run-length*, *symbol*)
  - For images, the maximum run-length is the size of a row
- Run length coding (RLC) is effective when long sequences of the same symbol occur.
- Run length coding exploits the spatial redundancy by coding the number of symbols in a run.

RUN LENGTH CODING

- The term run is used to indicate the repetition of a Symbol, while the term run length is used to represent the number of repeated symbols.
- Run length coding maps a sequence of numbers into a sequence of symbol pairs (run value).
- Images with large areas of constant shade are good candidates for this kind of compression.
- It is used in windows bitmap file format.
- Run length coding can be classified into

I) 1-D run length coding.

II) 2-D run length coding.

RUN LENGTH CODING

I) I-D run length coding-

In 1-D run length coding, each scan line is encoded independently.

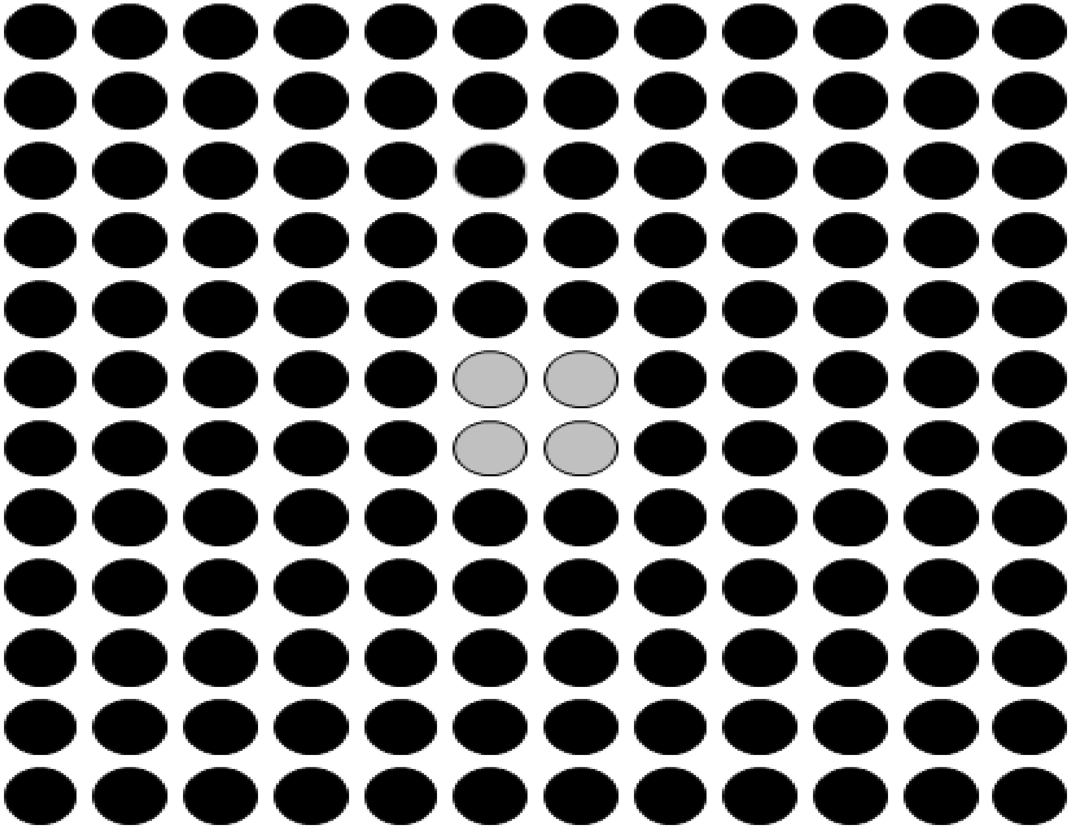
1-D RLC utilizes only the horizontal correlationship between pixels on the same scan line.

2-D RLC utilizes both horizontal and vertical correlationship between pixels.

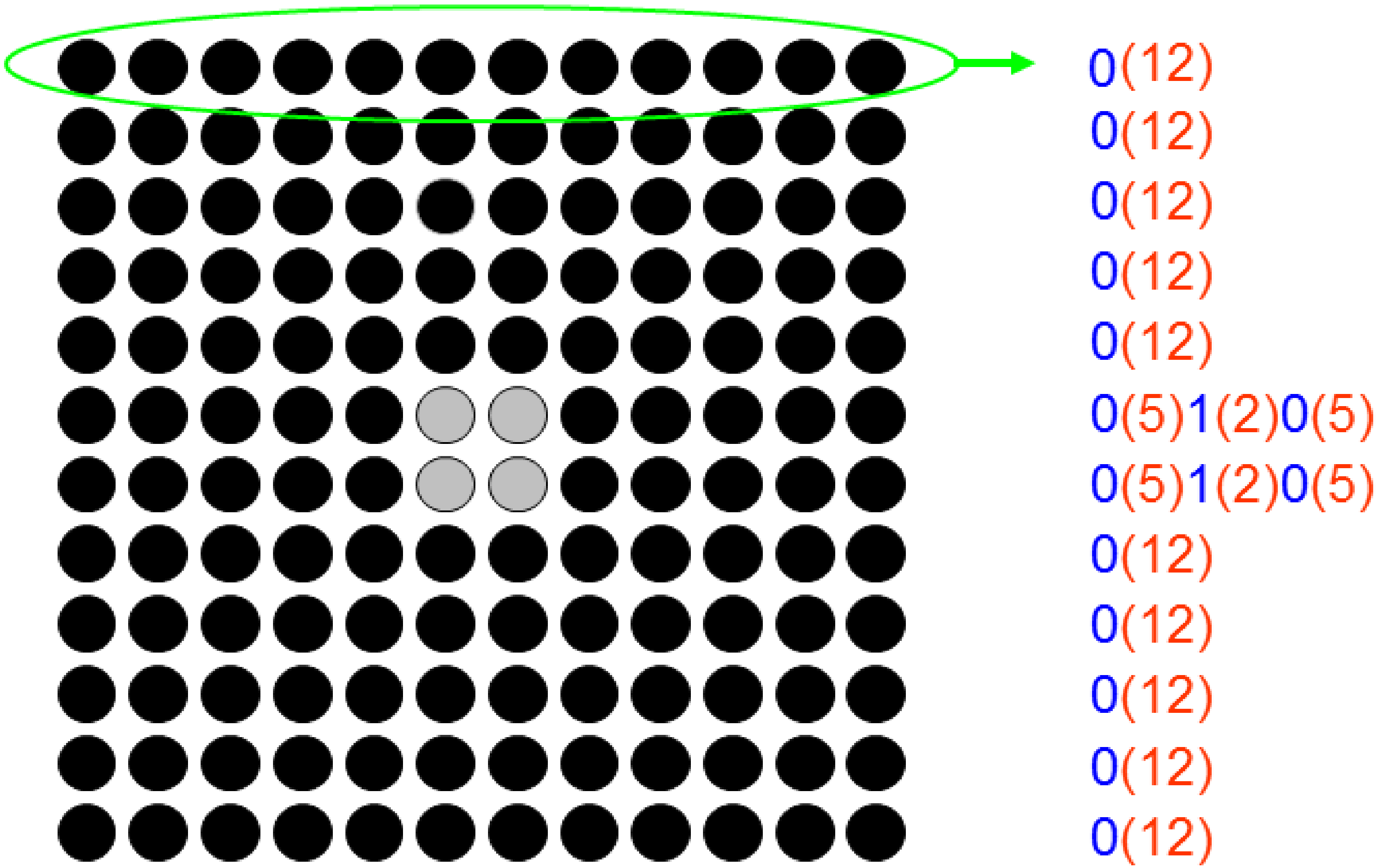
RUN LENGTH CODING

- 2-D run length coding
- The 1-D run length coding utilizes the correlation between pixels with in a scanline.
- In order to utilize correlation between pixels in neighbouring scan lines to achieve higher coding efficiency, 2-D run length coding was developed.
- In RLC, two values are transmitted- the first value indicates the number of times a particular symbol has occurred. The second value indicates the actual symbol.

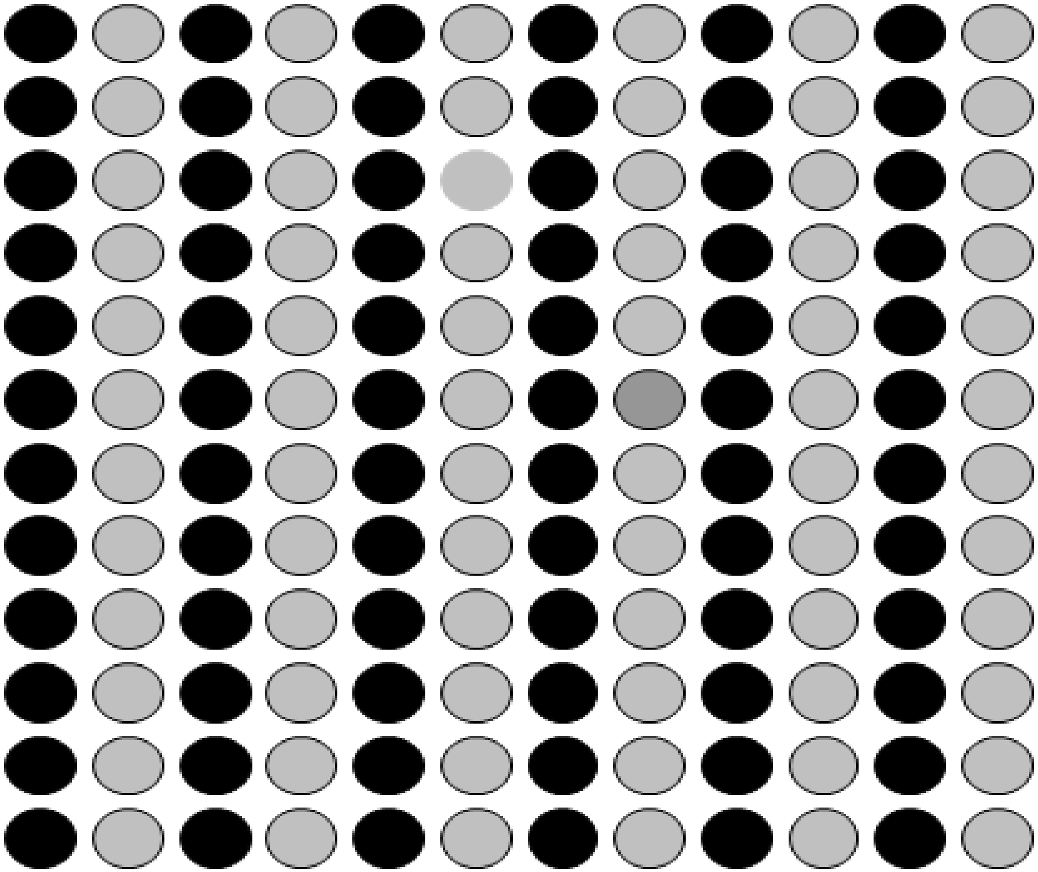
RUN LENGTH CODING



RUN LENGTH CODING



RUN LENGTH LIMITATIONS



ENTROPY

- The average bit rate of a coding scheme is given by:

$$\bar{b} = \sum_{i=1}^{L} p_i \, b_i$$

- p_i = probability of occurrence of the i^{th} reconstruction level
- b_i = number of bits assigned to the i^{th} message

Entropy

- In information theory, **entropy** is a measure of the uncertainty associated with a random variable.
- This term usually refers to the **Shannon entropy**, which quantifies the expected value of the information contained in a message.
- Entropy is typically measured in bits.
- Entropy, in an information sense, is a measure of unpredictability.
- Compressed message is more unpredictable, because there is no redundancy; each bit of the message is communicating a unique bit of information.
- Entropy is a measure of how much information the message contains.

ENTROPY

E = entropy(I)

E=a scalar value representing the entropy of grayscale image I. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = \sum_{i=1}^{n} p(x_i) \log_b \frac{1}{p(x_i)} = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i),$$

where p contains the histogram counts returned from imhist.
where b is the base of the logarithm used. Common values of b are 2,10.

A SIMPLE EXAMPLE

- Suppose we have a message consisting of 5 symbols, e.g. [►♣♣♠☻►♣☼► ☻]
- How can we code this message using 0/1 so the coded message will have minimum length (for transmission or saving!)

►	000
♣	001
☻	010
♠	011
☼	100

- 5 symbols → at least 3 bits
- For a simple encoding,

  length of code is 10*3=30 bits

A SIMPLE EXAMPLE – CONT.

- Intuition: Those symbols that are more frequent should have smaller codes, yet since their length is not the same, there must be a way of distinguishing each code

- For Huffman code,

length of encoded message

will be ►♣♣♠☻►♣☼► ☻

=3*2 +3*2+2*2+3+3=24bits

Symbol	Freq.	Code
►	3	00
♣	3	01
☻	2	10
♠	1	110
☼	1	111

ENTROPY CODING

Example 7.3 Calculate the entropy for the symbols shown in Table 7.2.

Table 7.2 Symbols and their distribution

Symbol	1	2	3	4	5	6
Probability	0.4	0.2	0.2	0.1	0.05	0.05

Solution Entropy $= -\Sigma p_i \times \log_2 p_i$; as $\log_2 x = \log_{10} x / \log_{10} 2$

$= -[0.4 \times (\log10(0.4)/\log10(2)) + 0.2 \times (\log10(0.2)/\log10(2)) + 0.2 \times (\log10(0.2)/\log10(2)) + 0.1 \times (\log10(0.1)/\log10(2)) + 0.05 \times (\log10(0.05)/\log10(2)) + 0.05 \times (\log10(0.05)/\log10(2))]$

$= -[-0.5288 - 0.4644 - 0.4644 - 0.3322 - 0.2161 - 0.2161]$

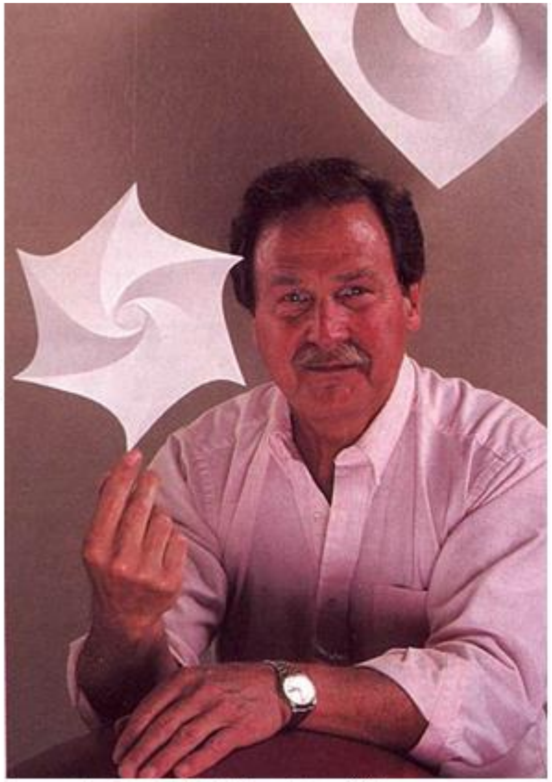
$= 2.22$

HUFFMAN CODE

- Developed by David A. Huffman in 1951
- Widely used in Computers, HDTV, modems etc
- Robert M. Fano assigned a term paper on efficient representation of numbers

Robert M. Fano

Claude Shannon



David A. Huffman

HUFFMAN CODING

- Huffman codes are optimal codes that map one symbol with one code word.
- In Huffman coding, it is assumed that each pixel intensity has associated with it a certain probability of occurrence, and this probability is spatially invariant.
- Huffman coding assigns a binary code to each intensity value, with shorter codes going to intensities with higher probability.
- If the probabilities can be estimated then the table of Huffman codes can be fixed in both the encoder and the decoder.

HUFFMAN CODING

- The Parameters involved in Huffman coding are as follows.
- **Entropy**
- **Average Length**
- **Efficiency**
- **Variance.**

- **Prefix Code-**A code is a prefix code if no code word is the prefix of another code word. The main advantage of a prefix code is that it is uniquely decodable.

  Example-Huffman code.

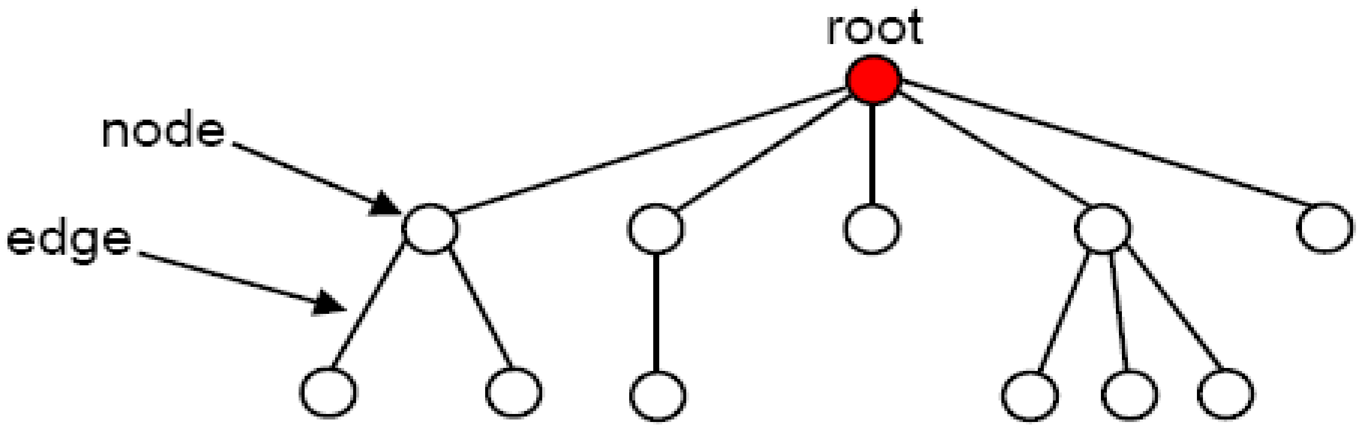
HUFFMAN CODING ALGORITHM

- Initialization: Put all symbols on a list sorted according to their frequency counts.
  - From the list pick two symbols with the lowest frequency counts. Form a Huffman subtree that has these two symbols as child nodes and create a parent node.
  - Assign the sum of the children's frequency counts to the parent and insert it into the list such that the order is maintained.
  - Delete the children from the list.
  - Repeat until the list has only one symbol left
- Assign a codeword for each leaf based on the path from the root.

Properties of Huffman Codes

- No Huffman code is the prefix of any other Huffman codes so decoding is unambiguous
- The Huffman coding technique is optimal (but we must know the probabilities of each symbol for this to be true)
- Symbols that occur more frequently have shorter Huffman codes

WHAT IS A TREE?



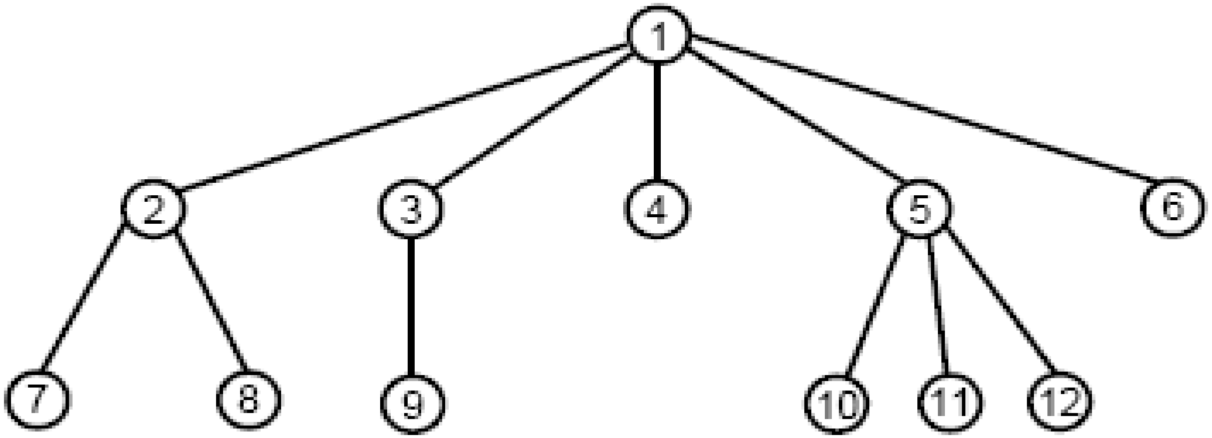
A tree consists of:

• a set of *nodes*

• a set of *edges*, each of which connects a pair of nodes.

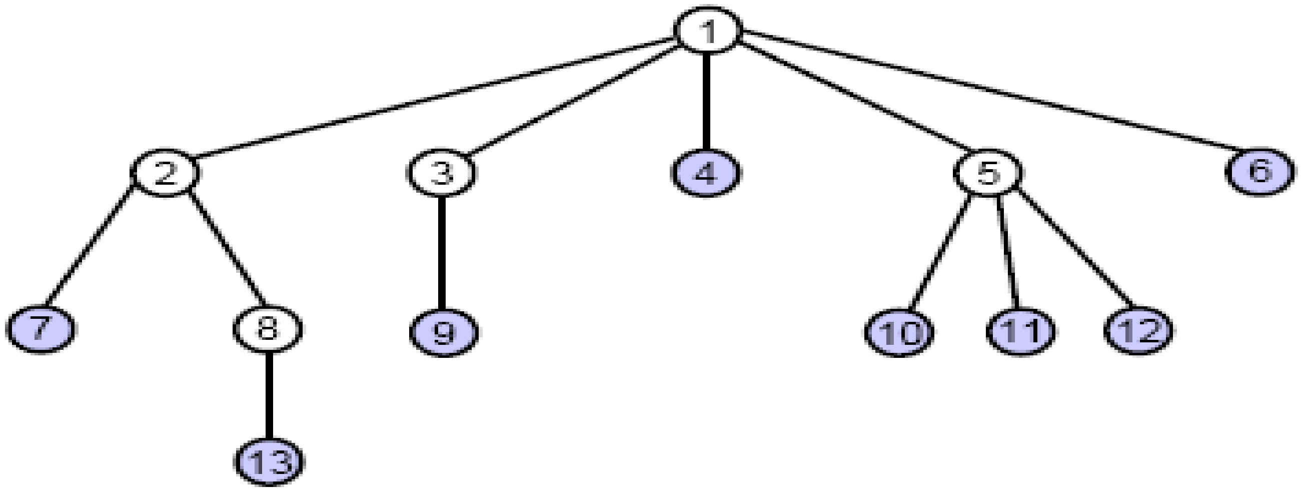
The node at the "top" of the tree is called the *root* of the tree.

Relationships Between Nodes



- If a node N is connected to other nodes that are directly below it in the tree, N is referred to as their *parent* and they are referred to as its *children*.
- example: node 5 is the parent of nodes 10, 11, and 12
- Each node is the child of *at most one* parent.

Types of Nodes



- A *leaf node* is a node without children.
- An *interior node* is a node with one or more children.

Example

Example 9.2 *Obtain the Huffman code for the word 'COMMITTEE'*

Solution Total number of symbols in the word **'COMMITTEE'** is 9.

$$\text{Probability of a symbol} = \frac{\text{Total number of occurrence of symbol in a message}}{\text{Total number of symbol in the message}}$$

Probability of the symbol C = $p(C) = 1/9$

Probability of the symbol O = $p(O) = 1/9$

Probability of the symbol M = $p(M) = 2/9$

Probability of the symbol I = $p(I) = 1/9$

Probability of the symbol T = $p(T) = 2/9$

Probability of the symbol E = $p(E) = 2/9$

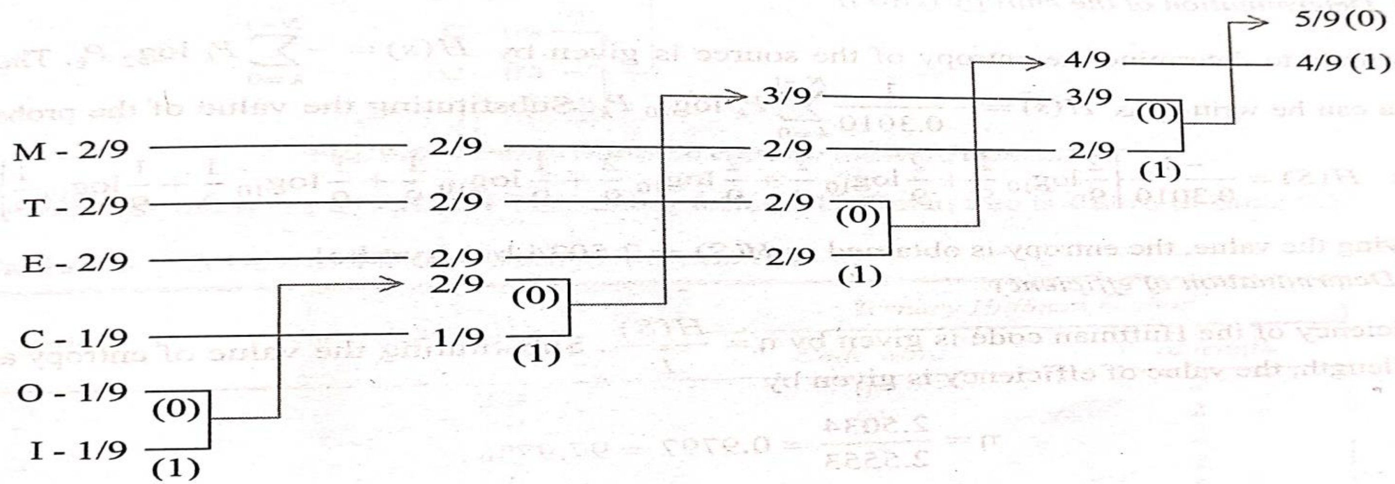
Step 1 Arrange the symbols into descending order according to the probability. This is illustrated in Table 9.1.

Table 9.1 *Arrangement of symbols in descending order of probability*

Symbol	*Probability*
M	2/9
T	2/9
E	2/9
C	1/9
O	1/9
I	1/9

Step 2 Construction of Huffman tree

The Huffman tree corresponding to the term COMMITTEE is shown below.



Step 3 Code word from the Huffman Tree

The code word for each symbol and the corresponding word length is shown in Table 9.2.

Table 9.2 *Code word from Huffman tree*

Symbol	*Probability*	*Binary Huffman method*	
		Codeword	*Word length*
M	2/9	01	2
T	2/9	10	2
E	2/9	11	2
C	1/9	001	3
O	1/9	0000	4
I	1/9	0001	4

$$H(S) = \frac{-1}{0.3010}\left\{\frac{2}{9}\log_{10}\frac{2}{9}+\frac{2}{9}\log_{10}\frac{2}{9}+\frac{2}{9}\log_{10}\frac{2}{9}+\frac{1}{9}\log_{10}\frac{1}{9}+\frac{1}{9}\log_{10}\frac{1}{9}+\frac{1}{9}\log_{10}\frac{1}{9}\right\}$$

$$H(S) = 2.5034 \text{ bits/symbol}$$

Finding the average length $\overline{L}$ (Ternary Huffman Coding)

$$\overline{L} = \sum_{k=0}^{N-1} P_k l_k = \frac{2}{9}\times 2+\frac{2}{9}\times 2+\frac{2}{9}\times 2+\frac{1}{9}\times 2+\frac{1}{9}\times 2+\frac{1}{9}\times 2$$

$$= 0.4444+0.4444+0.4444+0.2222+0.2222+0.2222$$

$$\overline{L} = 1.9998 \text{ bits/symbol}$$

***Efficiency of the compression in ternary Huffman Coding* (η)**

The efficiency of the Huffman code is given by

$$\eta = \frac{H(S)}{\overline{L}}$$

By substituting the value of $H(S)$, $\overline{L}$ we get

$$\eta = \frac{2.5034}{1.9998} = 125.18\%$$

$$\eta = 1.2518$$

HUFFMAN CODING EXAMPLE

Input Image

$$\begin{bmatrix} 1 & 2 & 5 & 7 \\ 2 & 3 & 7 & 5 \\ 7 & 2 & 1 & 3 \\ 6 & 4 & 7 & 1 \end{bmatrix}$$

STEP: 1

Probability of occurrence of gray level

Probability of '1' = P(1) = 3/16

Probability of '2' = P(2) = 3/16

Probability of '3' = P(3) = 2/16

Probability of '4' = P(4) = 1/16

Probability of '5' = P(5) = 2/16

Probability of '6' = P(6) = 1/16

Probability of '7' = P(7) = 4/16

Huffman Coding Example

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
	7	4/16					
							12

HUFFMAN CODING EXAMPLE

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
	7	4/16					
	1	3/16					
	2	3/16					
	3	2/16					
	5	2/16					
	4	1/16					
	6	1/16					

HUFFMAN CODING EXAMPLE

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
(01)	7	4/16	4/16	4/16	5/16	7/16	9/16 (0)
(11)	1	3/16	3/16	4/16	4/16	5/16 (0)	7/16 (1)
(000)	2	3/16	3/16	3/16	4/16 (0)	4/16 (1)	
(001)	3	2/16	2/16	3/16 (0)	3/16 (1)		
(100)	5	2/16	2/16 (0)	2/16 (1)			
(1010)	4	1/16 (0)	2/16 (1)				
(1011)	6	1/16 (1)					

Huffman coding Example

Pixel Value (or) Symbol	1	2	3	4	5	6	7
Probability, P_k	3/16	3/16	2/16	1/16	2/16	1/16	4/16
Code	11	000	001	1010	100	1011	01

HUFFMAN CODING EXAMPLE

STEP: 3 To find the Average length

$$\bar{L} = P_1 l_1 + P_2 l_2 + P_3 l_3 + P_4 l_4 + P_5 l_5 + P_6 l_6 + P_7 l_7$$

$$= \frac{3}{16}\times 2 + \frac{3}{16}\times 3 + \frac{2}{16}\times 3 + \frac{1}{16}\times 4 + \frac{2}{16}\times 3 + \frac{1}{16}\times 4 + \frac{4}{16}\times 2$$

$$= \frac{1}{16}[6+9+6+4+6+4+8]$$

$$= \frac{43}{16} = 2.6875$$

Where P_k is Probability of the symbol or pixel value

l_k is length of the code for the corresponding symbol

$\bar{L}$ is Average length

STEP: 4 To find the Entropy, H(s)

$$H(s) = -\sum_k P_k \log_2^{(P_k)}$$

$$H(s) = -\left\{\frac{3}{16}\times\log_2^{\left(\frac{3}{16}\right)} + \frac{3}{16}\times\log_2^{\left(\frac{3}{16}\right)} + \frac{2}{16}\times\log_2^{\left(\frac{2}{16}\right)} + \frac{1}{16}\times\log_2^{\left(\frac{1}{16}\right)} + \frac{2}{16}\times\log_2^{\left(\frac{2}{16}\right)} + \frac{1}{16}\times\log_2^{\left(\frac{1}{16}\right)} + \frac{4}{16}\times\log_2^{\left(\frac{4}{16}\right)}\right\}$$

$$= -\left\{\frac{3}{16}\times(-2.4153) + \frac{3}{16}\times(-2.4153) + \frac{2}{16}\times(-3.0003) + \frac{1}{16}\times(-4.0004) + \frac{2}{16}\times(-3.0003) + \frac{1}{16}\times(-4.0004) + \frac{4}{16}\times(-2)\right\}$$

$$= \frac{1}{16}\{7.2459 + 7.2459 + 6.0006 + 4.0004 + 6.0006 + 4.0004 + 8\}$$

$$= \frac{1}{16}\times 42.4938 = 2.6559$$

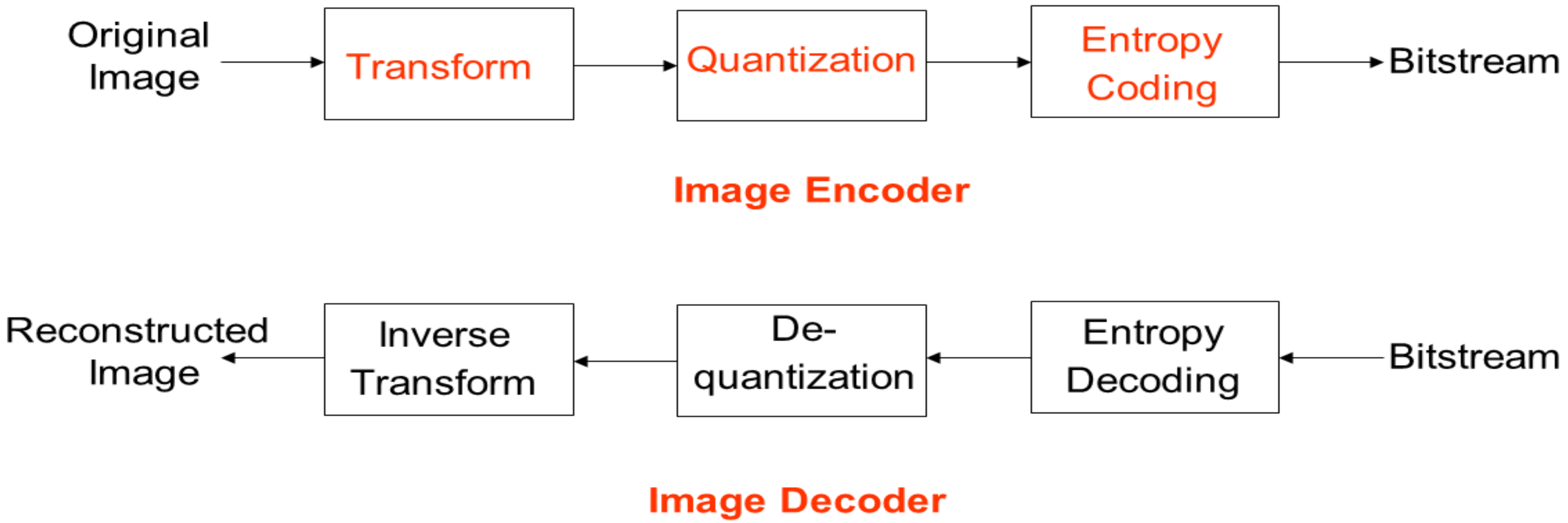
STEP: 5 To find the Efficiency $\eta = \frac{H(s)}{\bar{L}}$

$$\eta = \frac{2.6559}{2.6875}$$

$$= 0.9882$$

Efficiency in Percentage =0.9882 x 100 = **98.82**%

Transform based Image Compression



Thanks

Image Segmentation

By,

Prof (Dr.) P. P. Ghadekar

Outline

- Introduction
- Discontinuity and Similarity
- Edge Detection
- LoG
- Canny Edge Detection
- Watershed Segmentation etc.

Image Segmentation

- Segmentation subdivides an image into its constituent parts or objects with respect to a particular application .
- Segmentation should stop when the objects of interest in an application have been isolated.
- The segmentation is based on measurements taken from the image and might be ***greylevel, colour, texture, depth or motion***

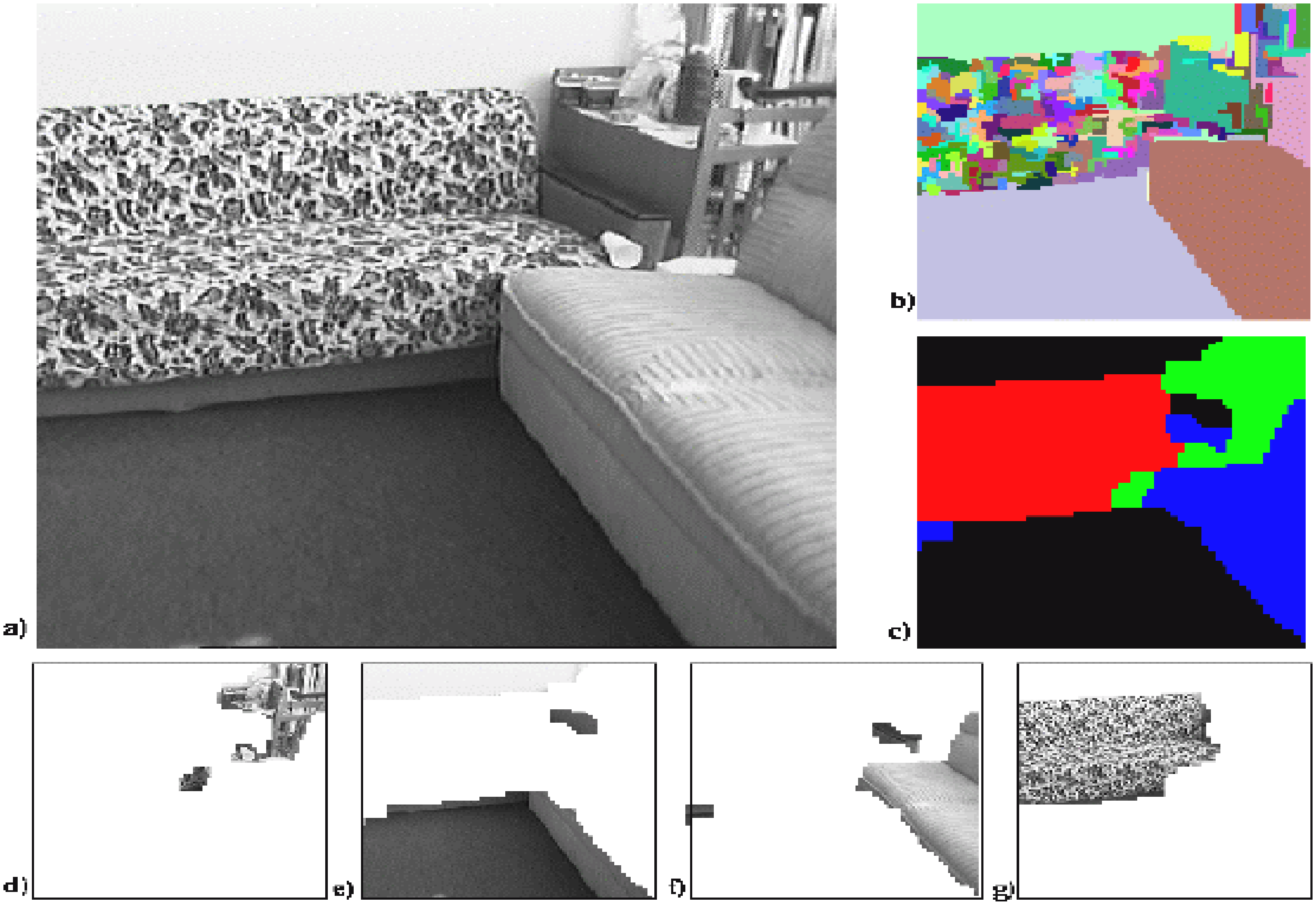
Introduction to Image Segmentation

- Example 1
  - Segmentation based on greyscale
  - Very simple 'model' of greyscale leads to inaccuracies in object labelling

Introduction to Image Segmentation

- Example 2
  - Segmentation based on texture
  - Enables object surfaces with varying patterns of grey to be segmented

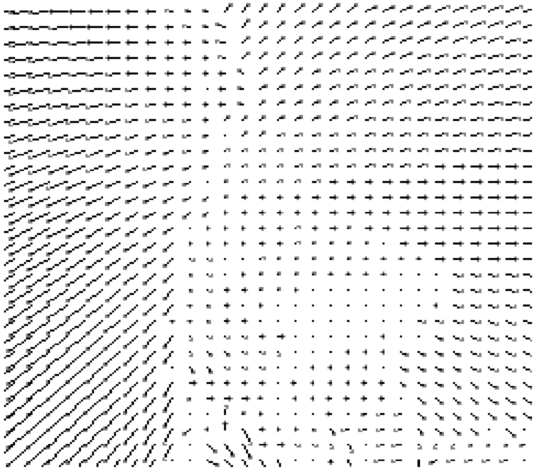
Introduction to Image Segmentation



Introduction to Image Segmentation

- Example 3
  - Segmentation based on motion
  - The main difficulty of motion segmentation is that an intermediate step is required to (either implicitly or explicitly) estimate an *optical flow field*
  - The segmentation must be based on this estimate and not, in general, the true flow

Introduction to Image Segmentation

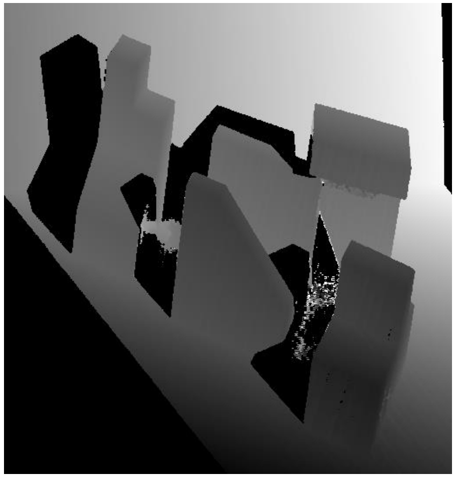


Introduction to Image Segmentation

- Example 3
  - Segmentation based on depth
  - This example shows **a range image**, obtained with **a laser range finder**
  - A segmentation based on the range (the object distance from the sensor) is useful **in guiding mobile robots**

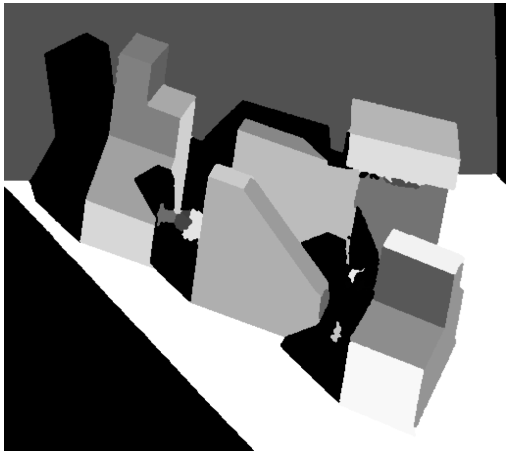
Introduction to Image Segmentation

Original image



Range image

Segmented image



The ***Range Image***, has pixel values that correspond to the distance.

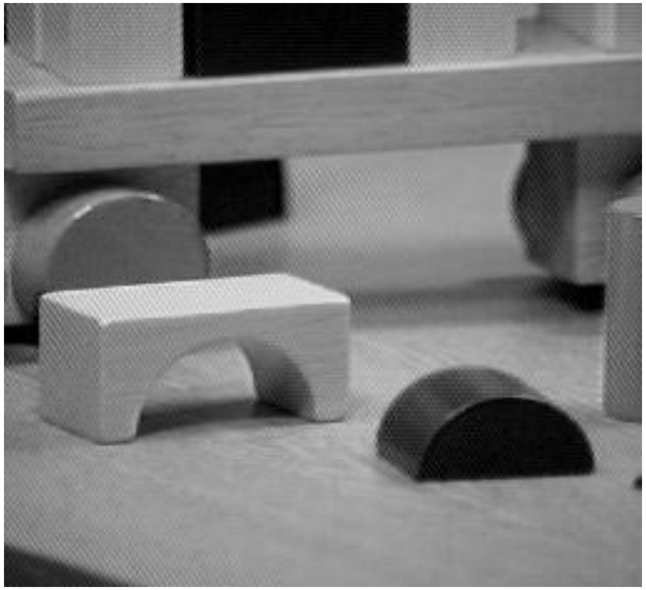
Image Segmentation

Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels.

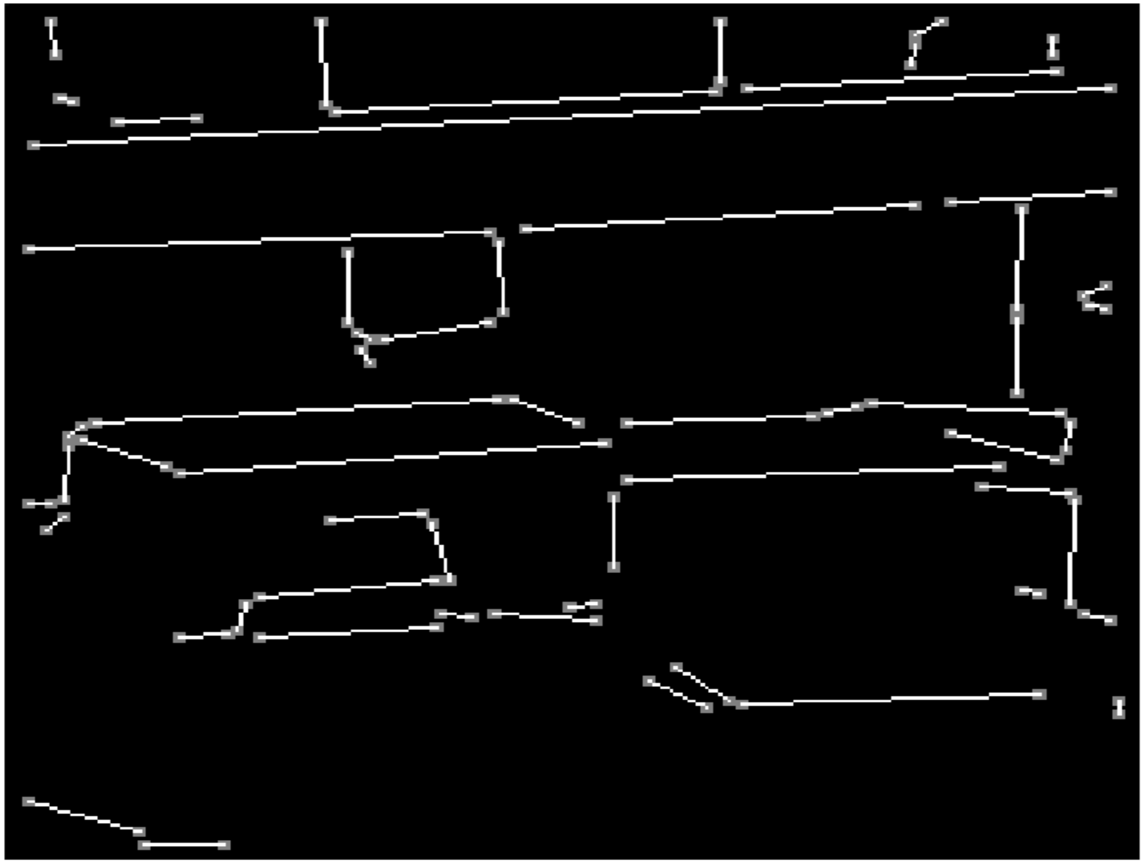
1. Into regions, which usually cover the image
2. Into linear structures, such as
   - line segments
   - curve segments
3. Into 2D shapes, such as
   - circles
   - ellipses
   - ribbons (long, symmetric regions)

Example 1: Regions

Example 2: Straight Lines



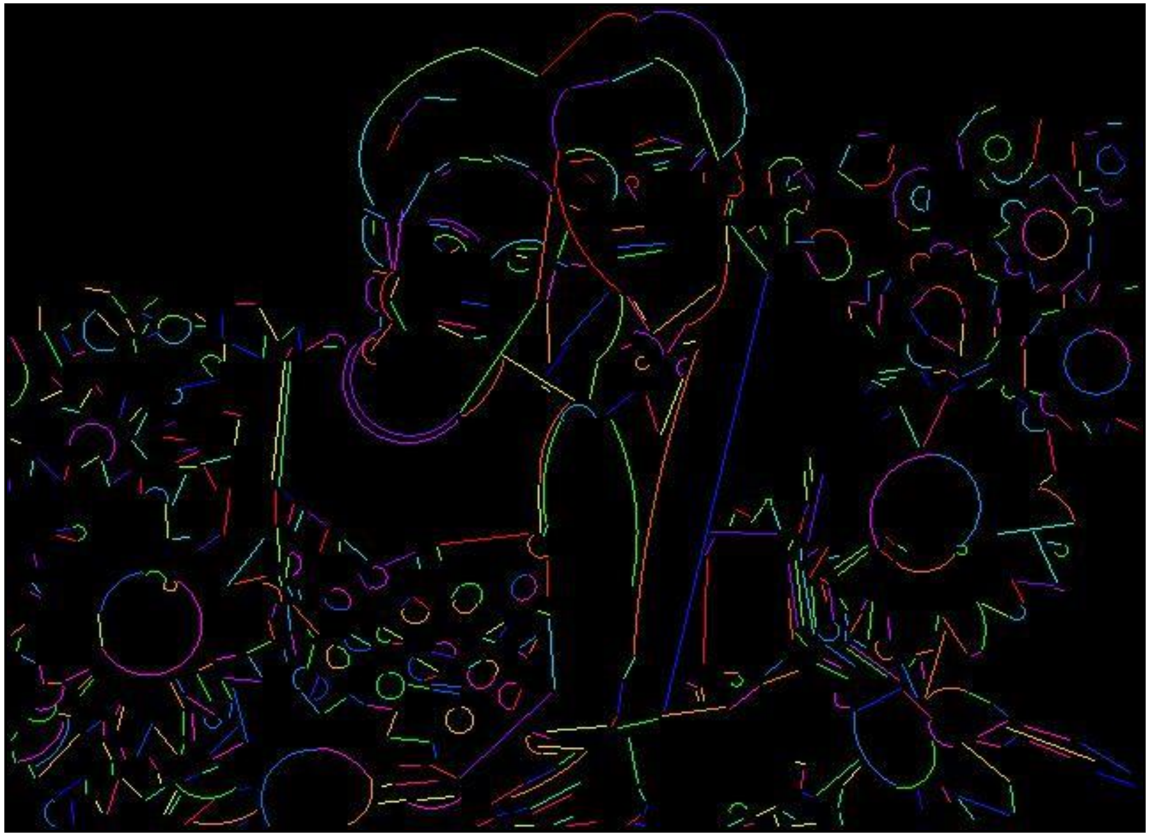
(a)



(b)

Example 3:Lines and Circular Arcs

(a)



(b)

Image Segmentation

- Segmentation algorithms for monochrome images are generally based on

1. Discontinuity
2. Similarity

Discontinuity

- ❖ In discontinuity, the approach is to partition an image based on the **abrupt changes** in gray level.
- ❖ The principal areas of detection in this approach are detection of **isolated points and detection of lines and edges** in an image.

Similarity

- In the other approach, the basis is mainly **thresholding, region growing, region splitting and merging.**

Detection of Discontinuities

- There are three basic types of grey level discontinuities that we tend to look for in digital images:
  - Points
  - Lines
  - Edges
- We typically find discontinuities using masks and correlation

Point Detection

- Mask used for detection of isolated point is –

  $R = W_1 Z_1 + W_2 Z_2 + \text{-----} + W_9 Z_9$

  where Z_i is the gray level of associated pixel

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Point Detection

Point detection can be achieved simply using the mask below:

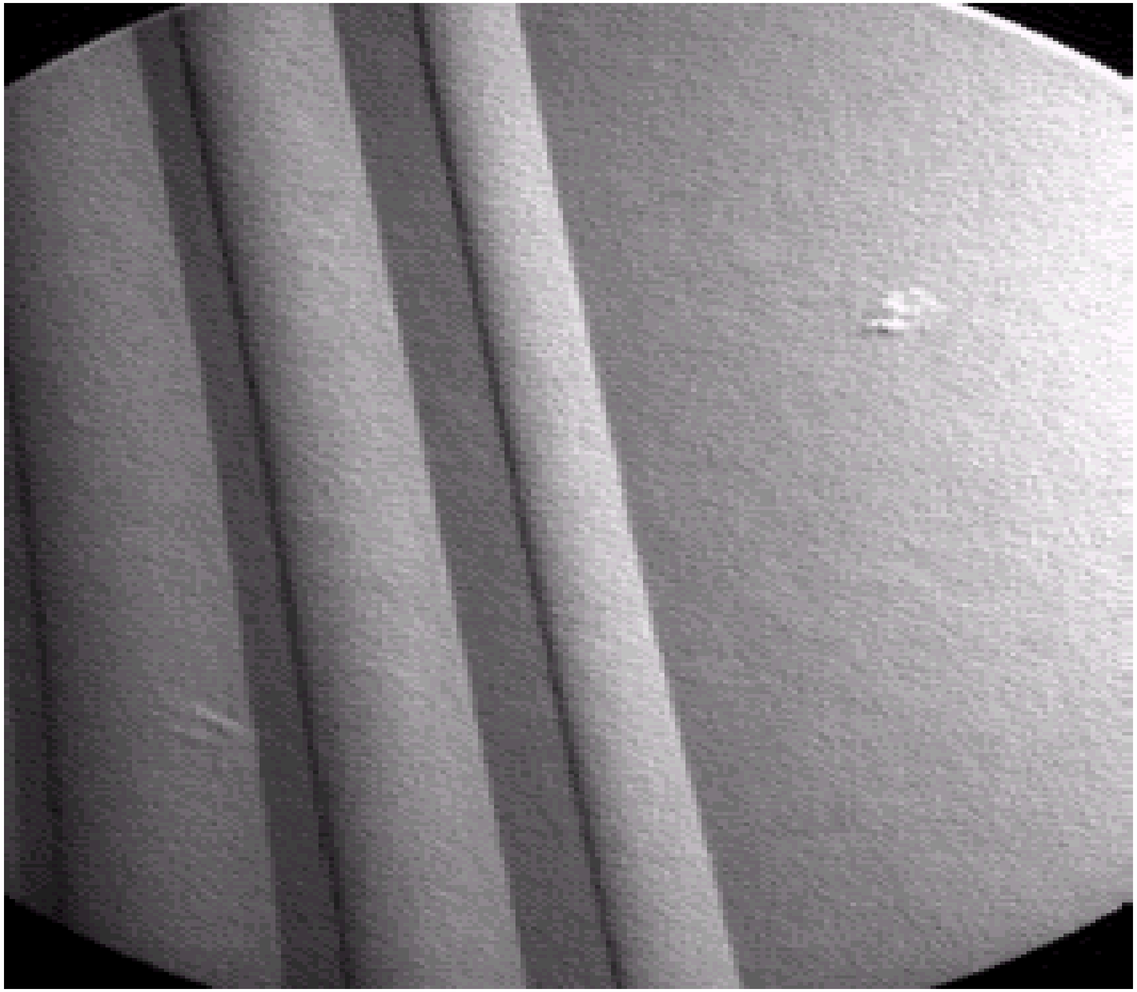
Points are detected at those pixels in the subsequent filtered image that are above a set threshold

−1	−1	−1
−1	8	−1
−1	−1	−1

Point Detection

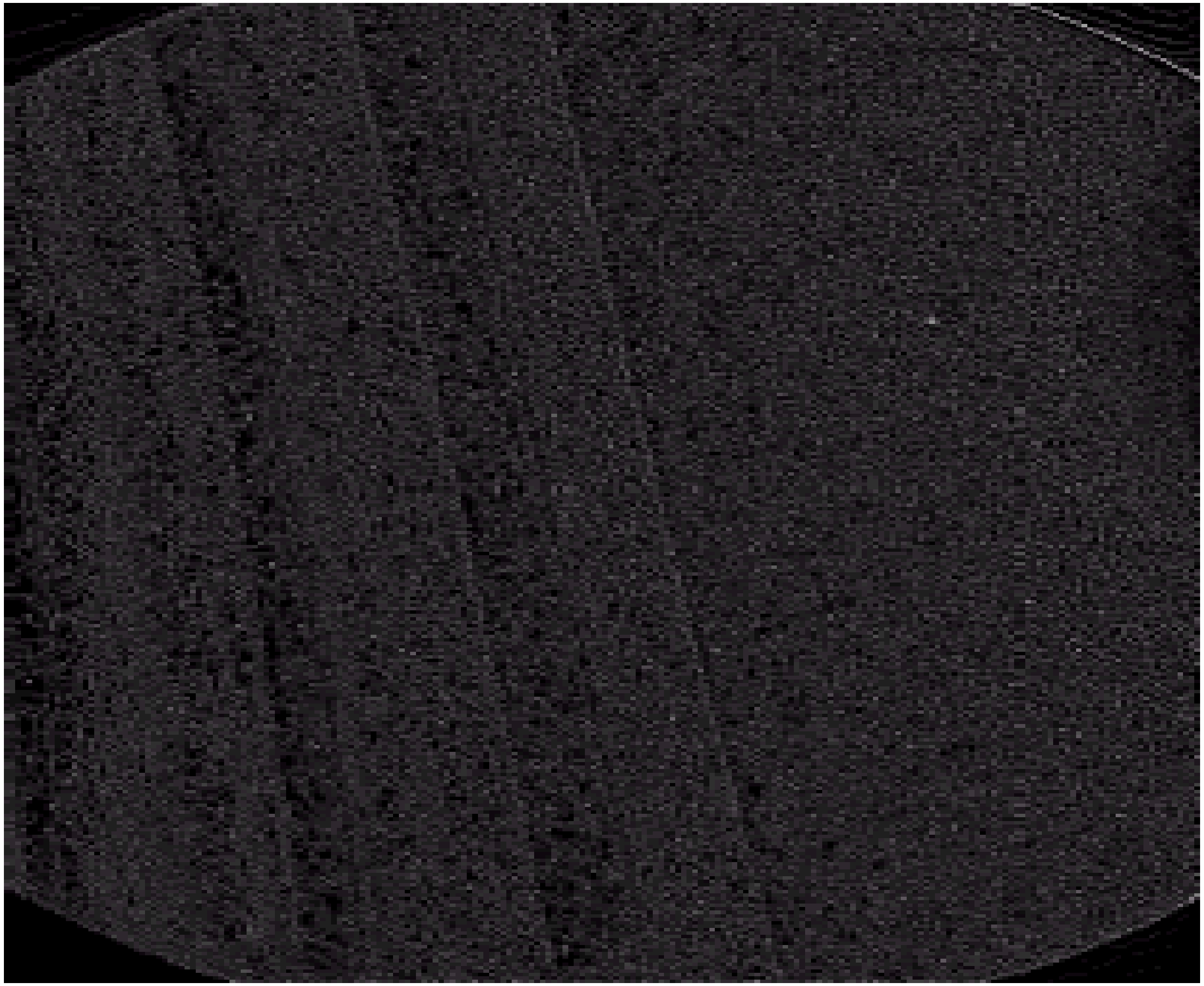
- It can be stated that a point has been detected at the location on which the mask has been centered if $|R| > T$, where T is a non negative threshold and R is as computed.
- Basically this formulation measures the weighted differences between the central point and its neighbors.
- It is based on the fact that the gray level of the isolated point will be quite different from the gray level of its neighbors

Point Detection



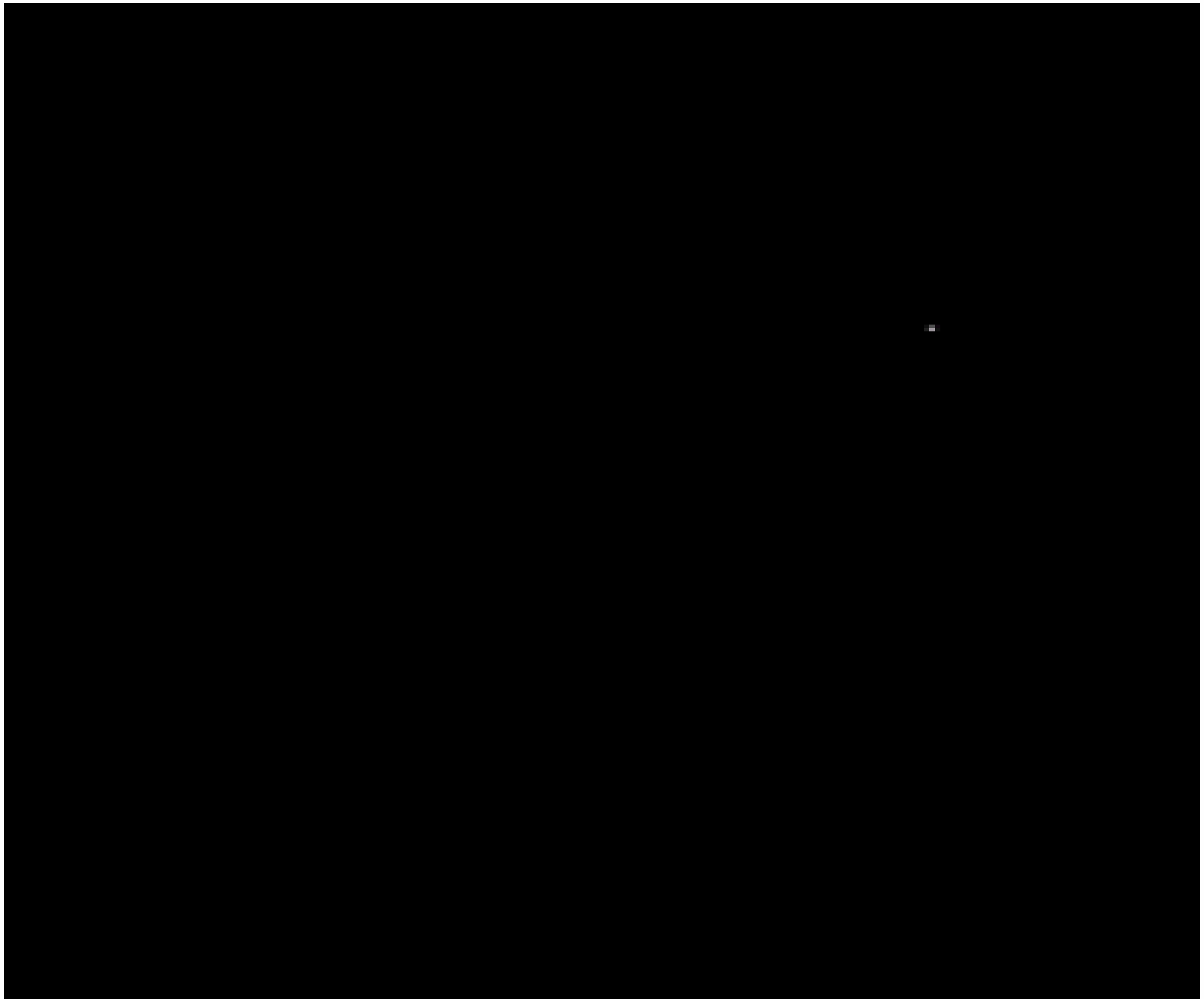
X-ray image of a turbine blade with porosity

Point Detection



Result of point detection

Point Detection



Line Detection

- Horizontal line detection

-1	-1	-1
2	2	2
-1	-1	-1

Line Detection

❖ + 45 ° line detection

-1	-1	2
-1	2	-1
2	-1	-1

Line Detection

- Vertical line detection

-1	2	-1
-1	2	-1
-1	2	-1

Line Detection

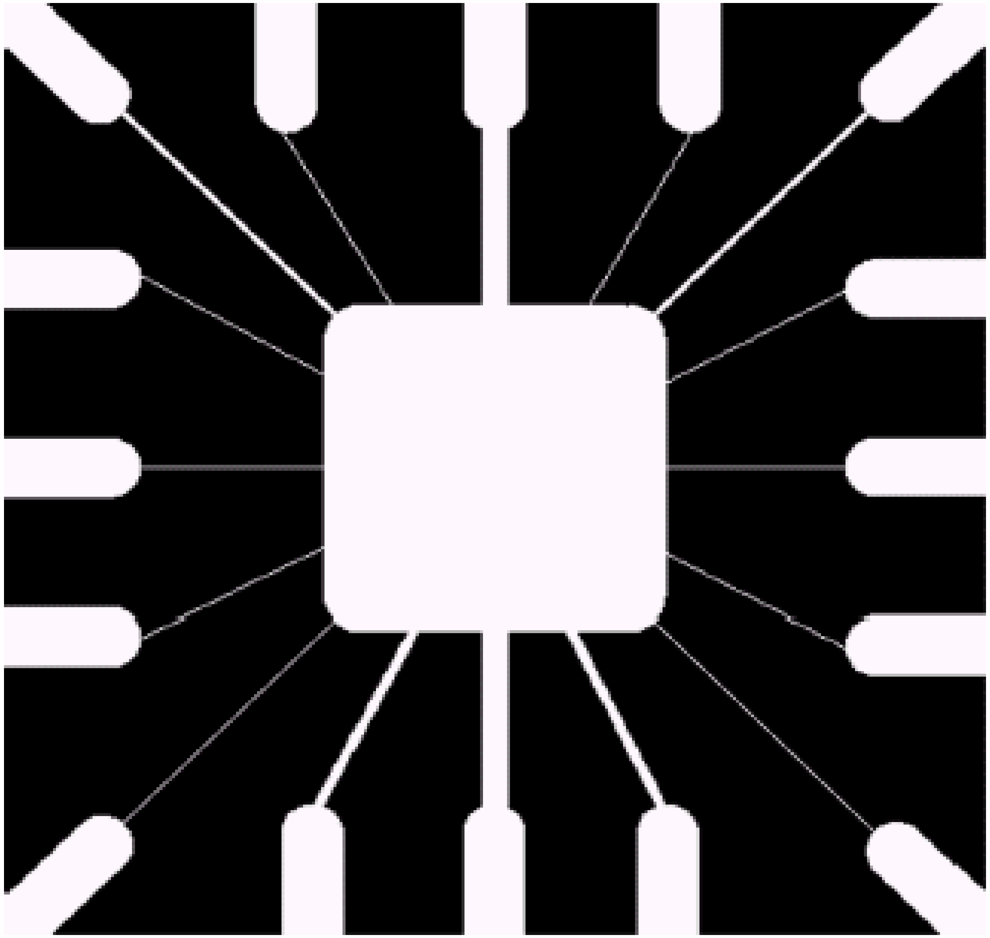
- - 45 ° line detection

2	-1	-1
-1	2	-1
-1	-1	2

Line Detection

- If **the first mask** were moved around the image it would respond more strongly to a pixel thick line oriented horizontally.
- If constant background is maintained / available, then maximum response would result when the line is passed through the middle row of the mask.

Line Detection



Binary Wire bond mask for an electronic circuit

Line Detection

Absolute value after applying -45 ° line detector

Line Detection

Result of Thresholding

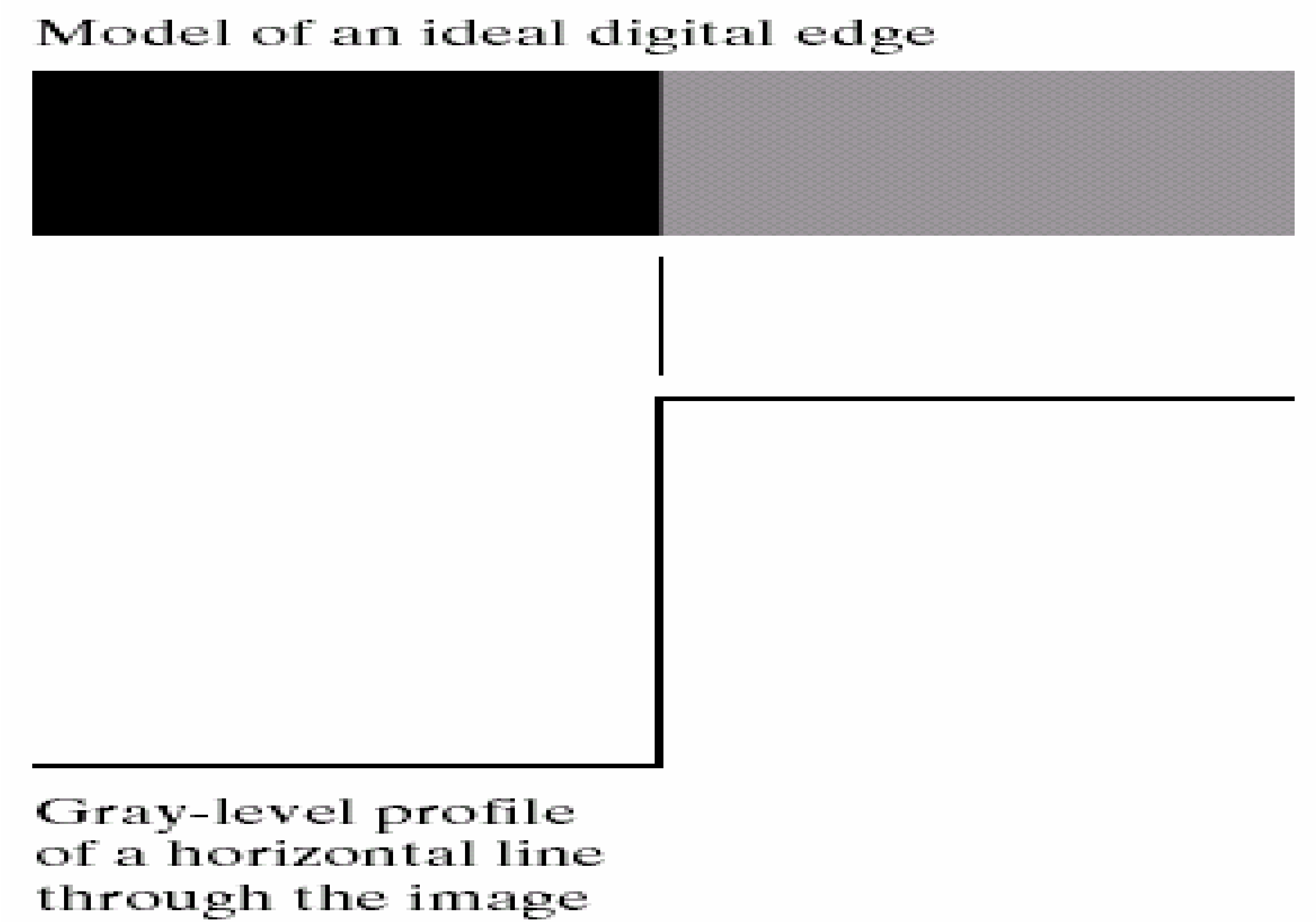
Detection of discontinuity

- The most common way to look for discontinuity is to run a mask through the image.

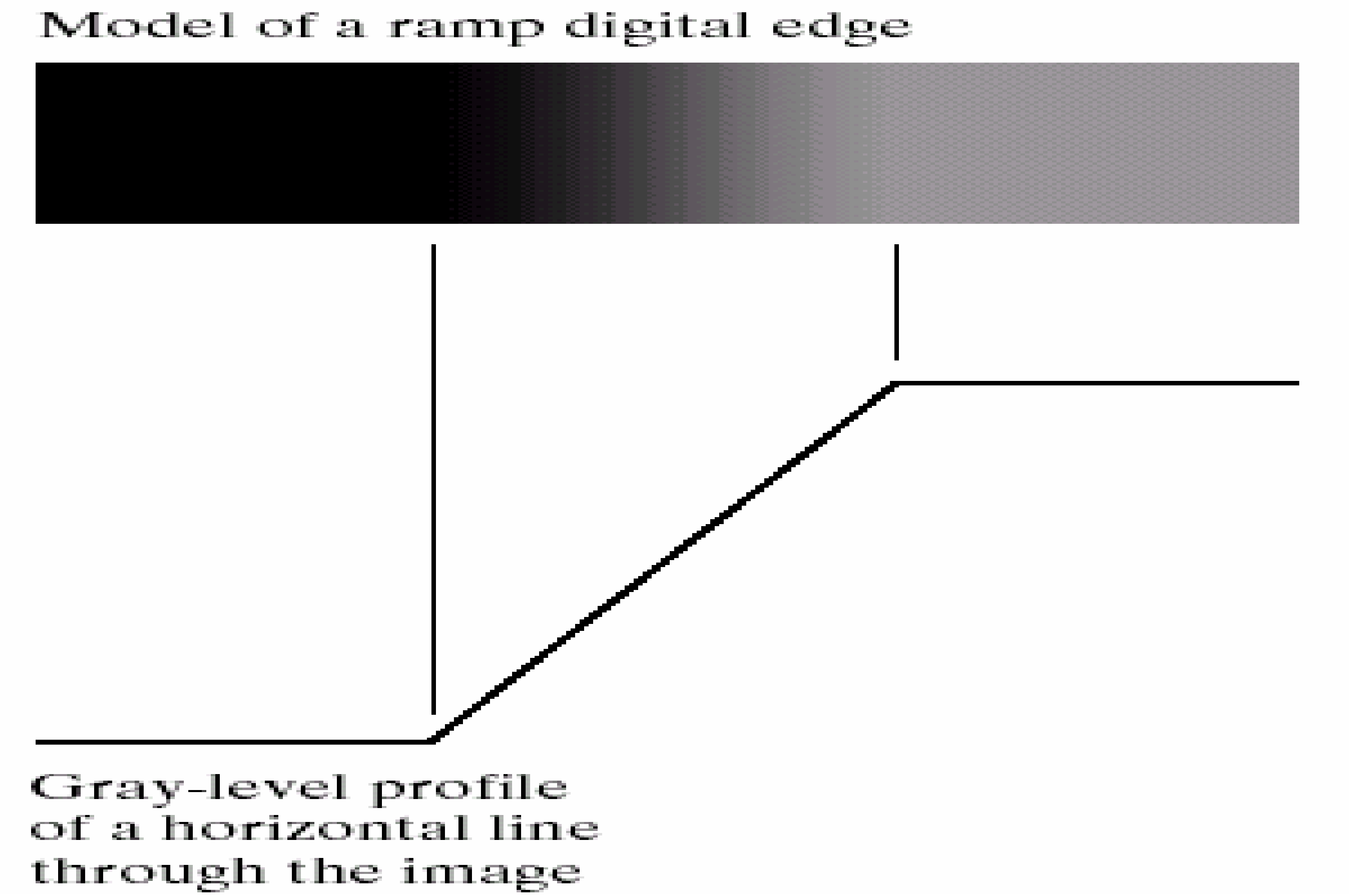
Edge Detection

- Edge detection is the approach of detecting meaningful discontinuities in gray level.
- It is because isolated points and thin lines are not frequent occurrences in most practical applications.

Edge Detection



Edge Detection

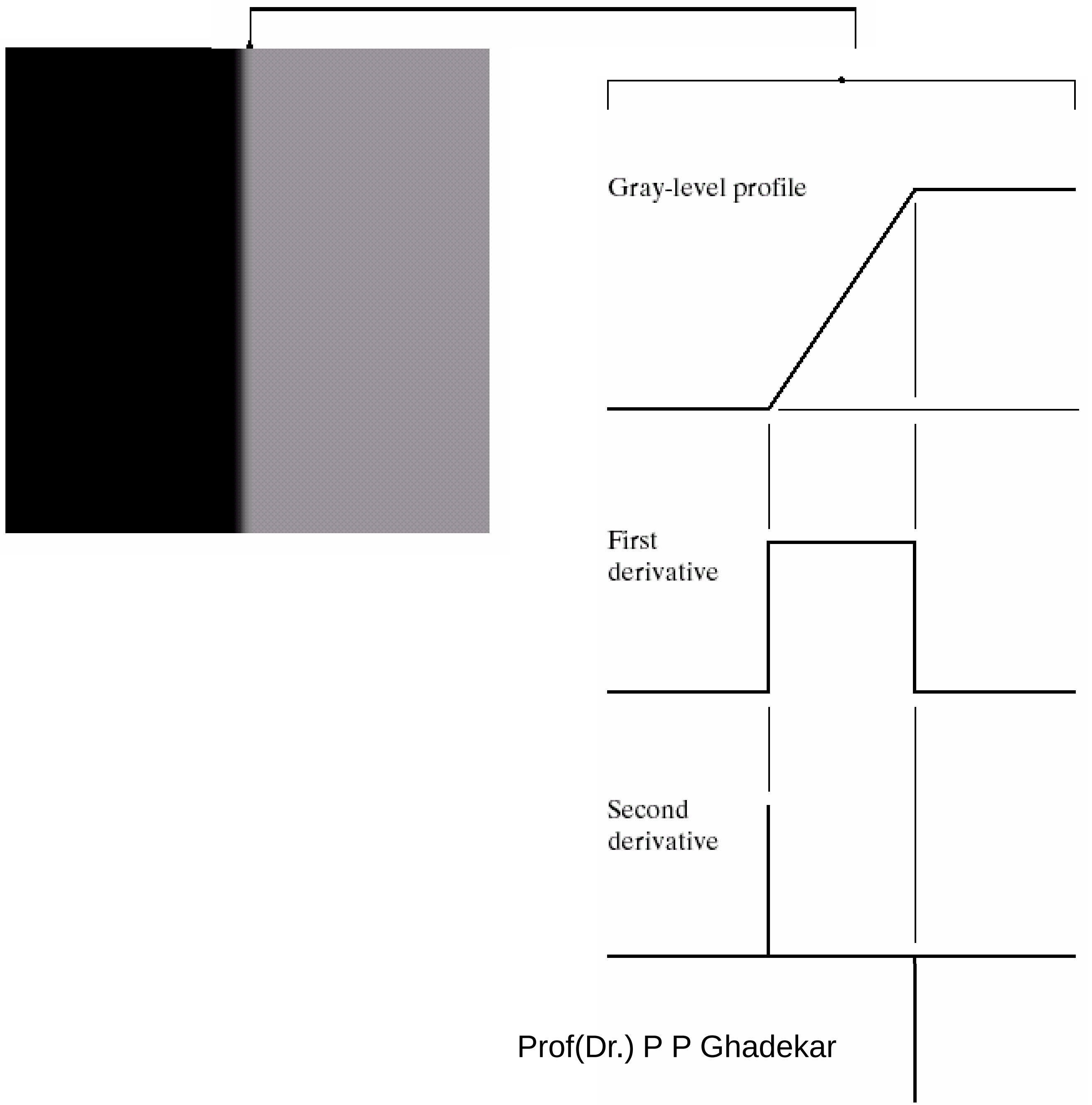


Edge Detection

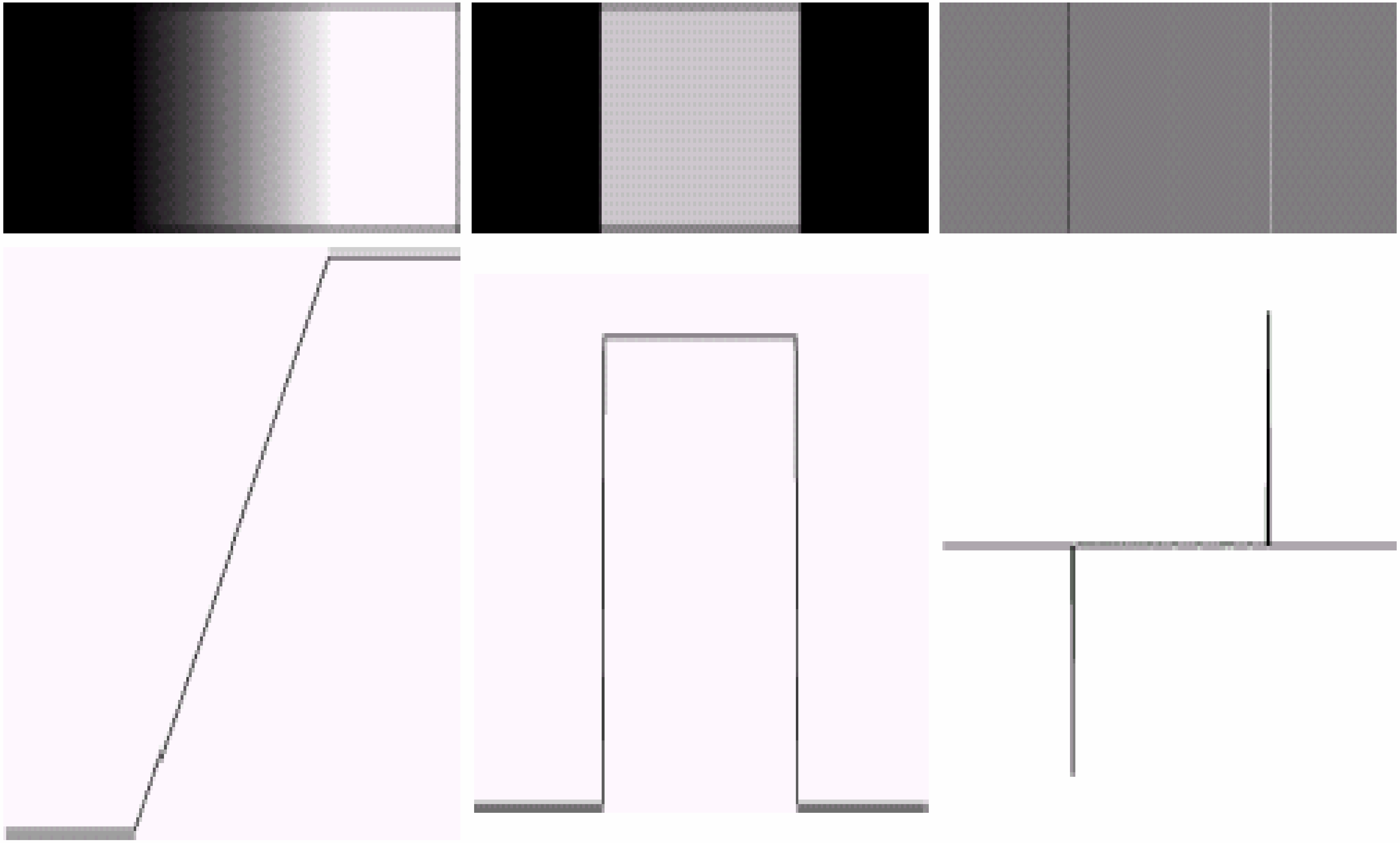
- An edge is the boundary between two regions with relatively distinct gray level properties.
- The regions are assumed to be sufficiently homogenous so that the transitions between the two regions can be determined on the basis of gray level discontinuities alone.
- The idea behind most effective edge detection techniques is the computation of a ‘local derivative’ operator.

Edge Detection

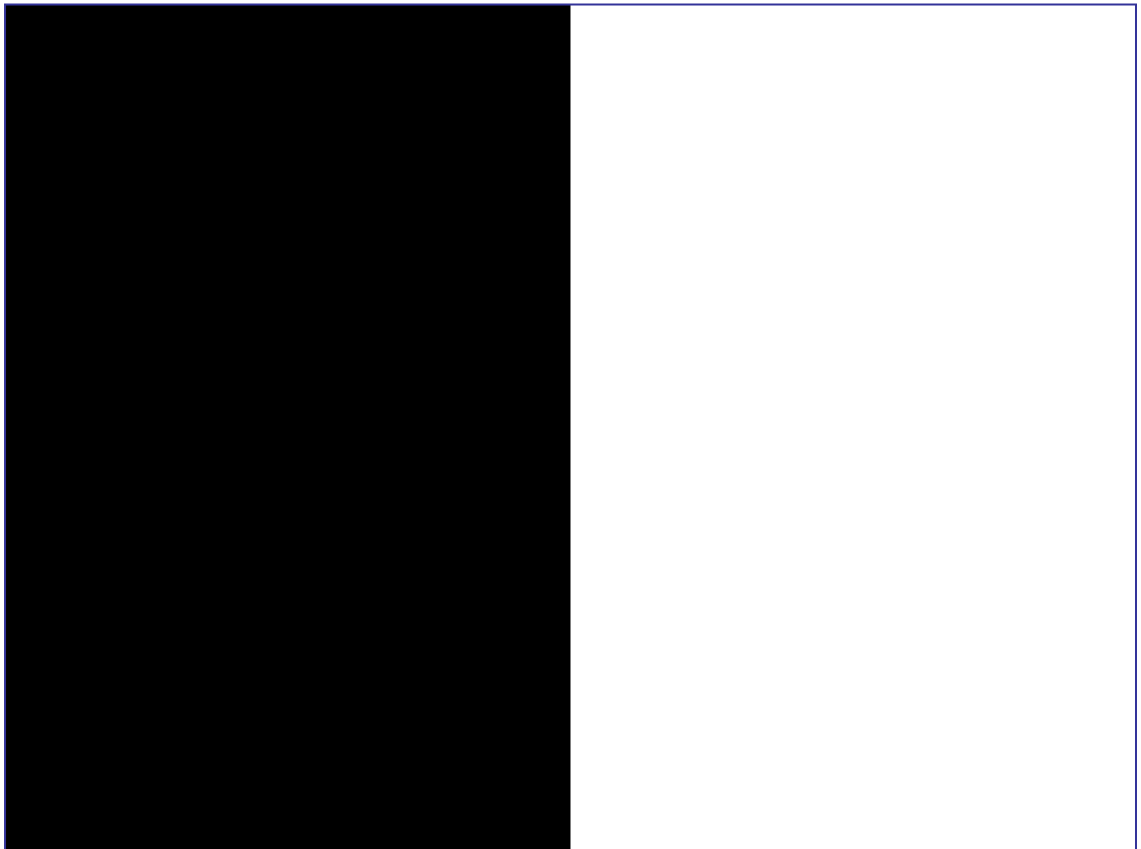
- The first derivative at any point of an image is obtained by using the magnitude of the gradient at that point.
- The second derivative is similarly obtained by using the Laplacian.


Gray-level profile
First
derivative
Second
derivative

Edge Detection



What is an Edge?



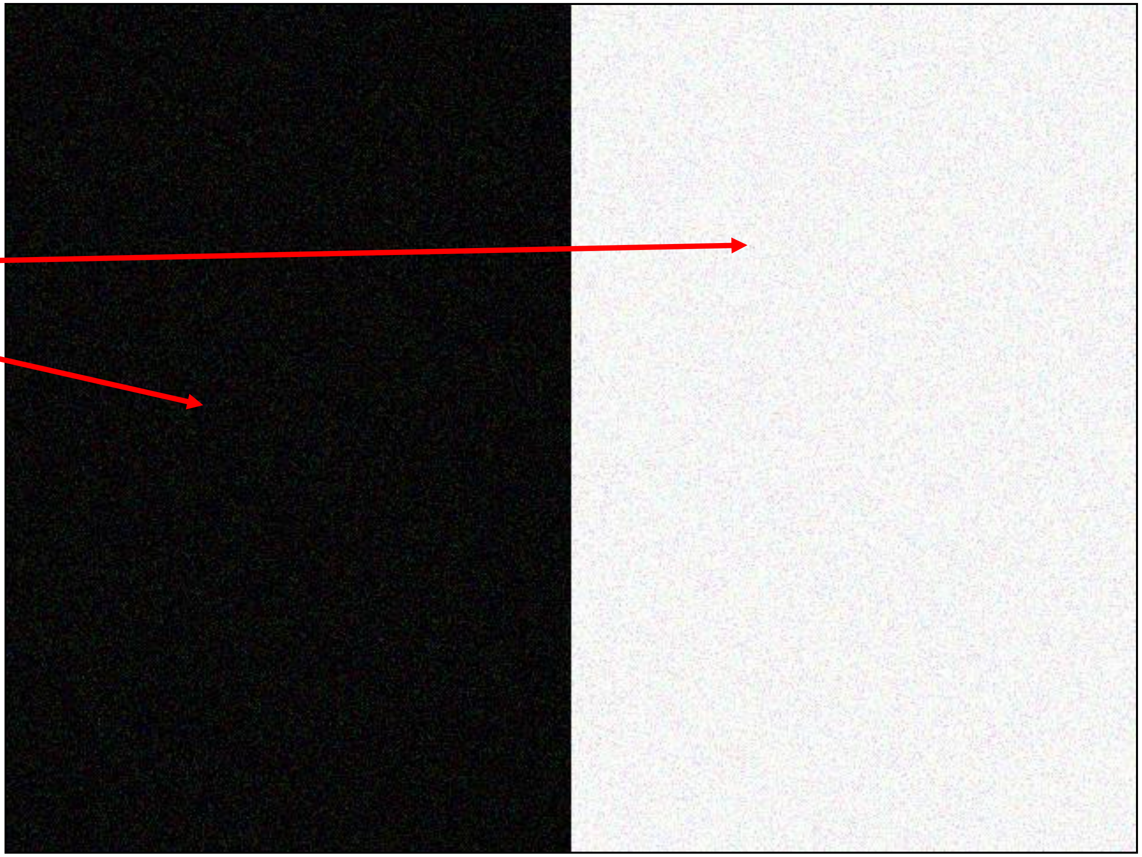
Edge easy to find

What is an Edge?

Where is edge? Single pixel wide or multiple pixels?

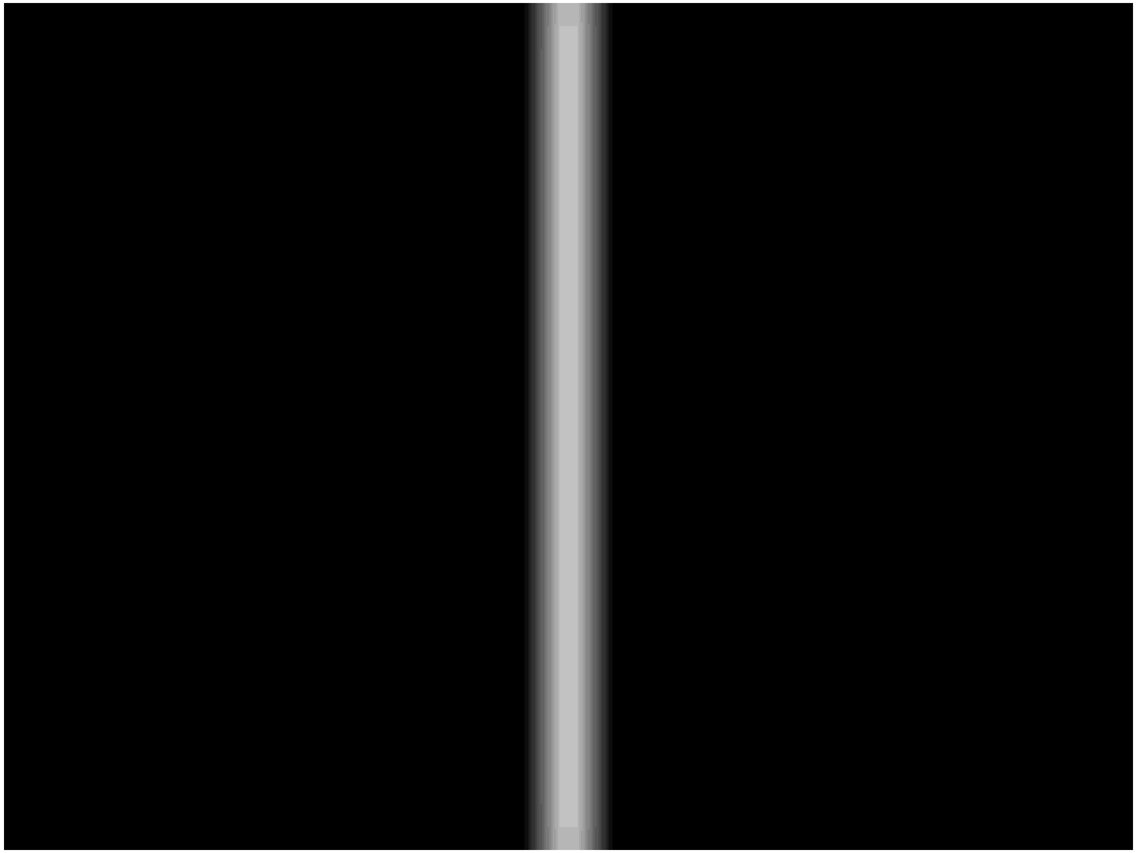
What is an Edge?

Noise is here



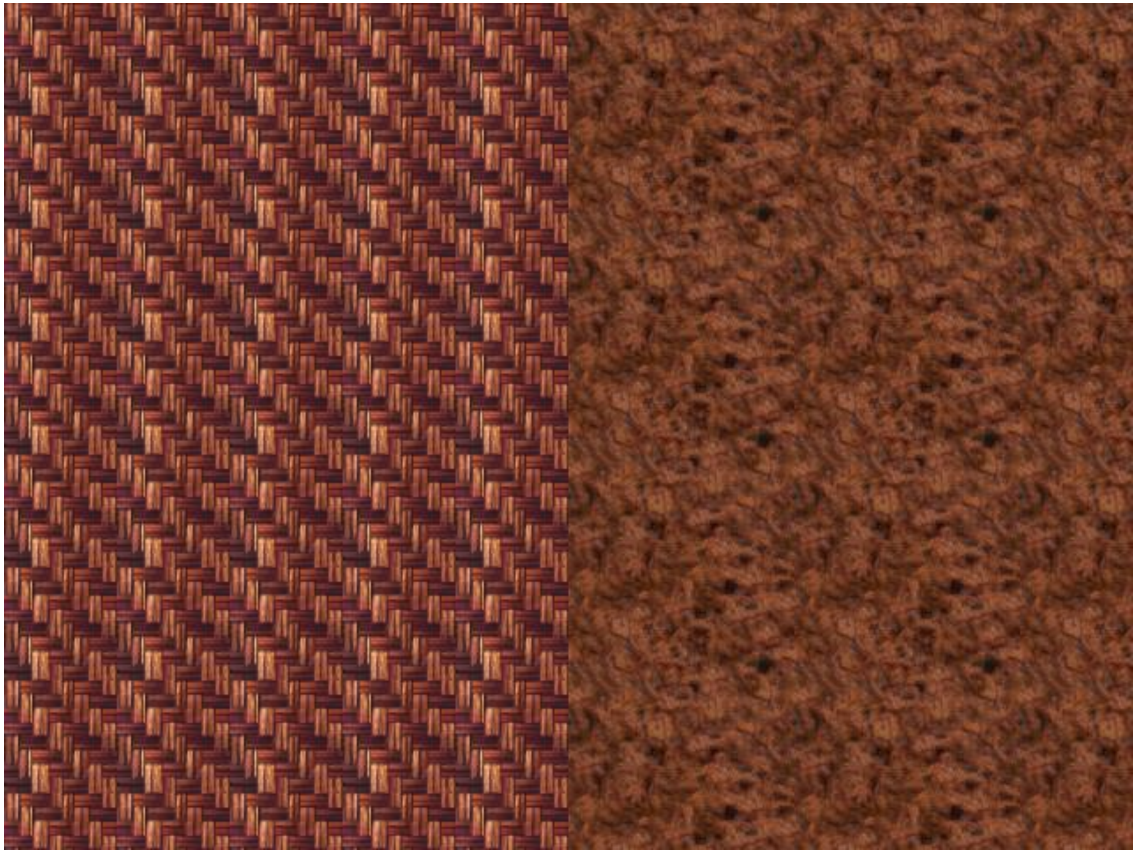
Noise: have to distinguish noise from actual edge

What is an Edge?



Is this one edge or two?

What is an Edge?



Texture discontinuity

First Derivative

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

First Derivative

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

First Derivative

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Laplacian Edge Detection

- We encountered the 2nd-order derivative based Laplacian filter already

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

- The Laplacian is typically not used by itself as it is too sensitive to noise.
- Usually when used for edge detection the Laplacian is combined with a smoothing Gaussian filter.

Laplacian

- The Laplacian of a 2D function f(x,y) is a second order derivative given as –

$$\mathbf{Del^2\, f = \partial^2 f / \partial x^2 + \partial^2 f / \partial x^2}$$

- For the 3 × 3 region, the two digital masks incorporated are -

0	−1	0
−1	4	−1
0	−1	0

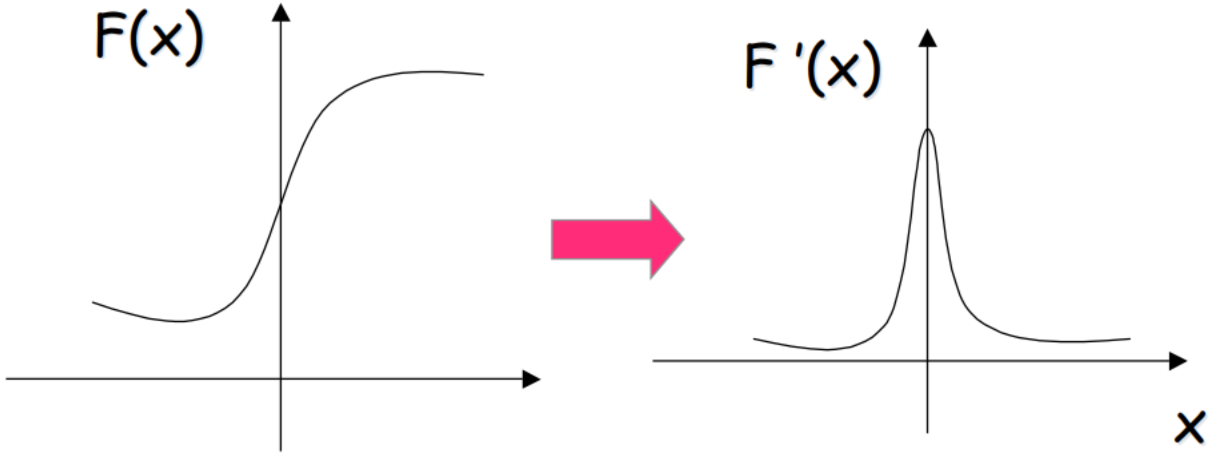
−1	−1	−1
−1	8	−1
−1	−1	−1

Laplacian

- The Laplacian is generally not used in its original form for edge detection because of its **unusual sensitivity to noise.**
- The **magnitude of Laplacian produces double edges, and it complicates the segmentation process.**
- Hence the general use of Laplacian is to find edges using its **Zero Crossing Property.**

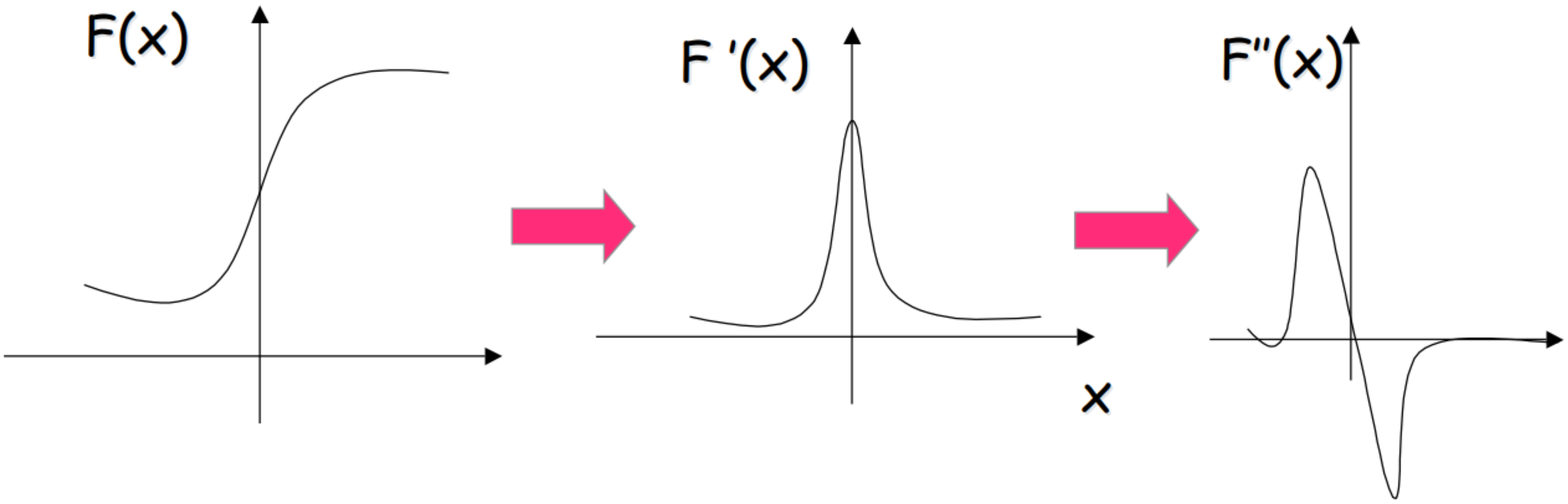
Laplacian

- Sharp changes in gray level of the input image correspond to **"peaks or valleys"** of the first-derivative of the input signal.



Laplacian

- Peaks or valleys of the first-derivative of the input signal, correspond to **“zero-crossings”** of the second-derivative of the input signal.



Laplacian-Zero Crossing Detector

- **The zero-crossing detector** looks for places in the Laplacian of an image where **the value of the Laplacian passes through zero.**
- **Points where the Laplacian Changes Sign.** Such points often occur at 'edges' in images.
- Points where the intensity of the image changes rapidly, but they also occur at places that are not as easy to associate with edges.

Laplacian

- Consider a 2-D Gaussian function,

$h(x,y) = -\exp(-(x^2 + y^2)/\sigma^2)$ where σ is the Standard Deviation

Let $r^2 = x^2 + y^2$

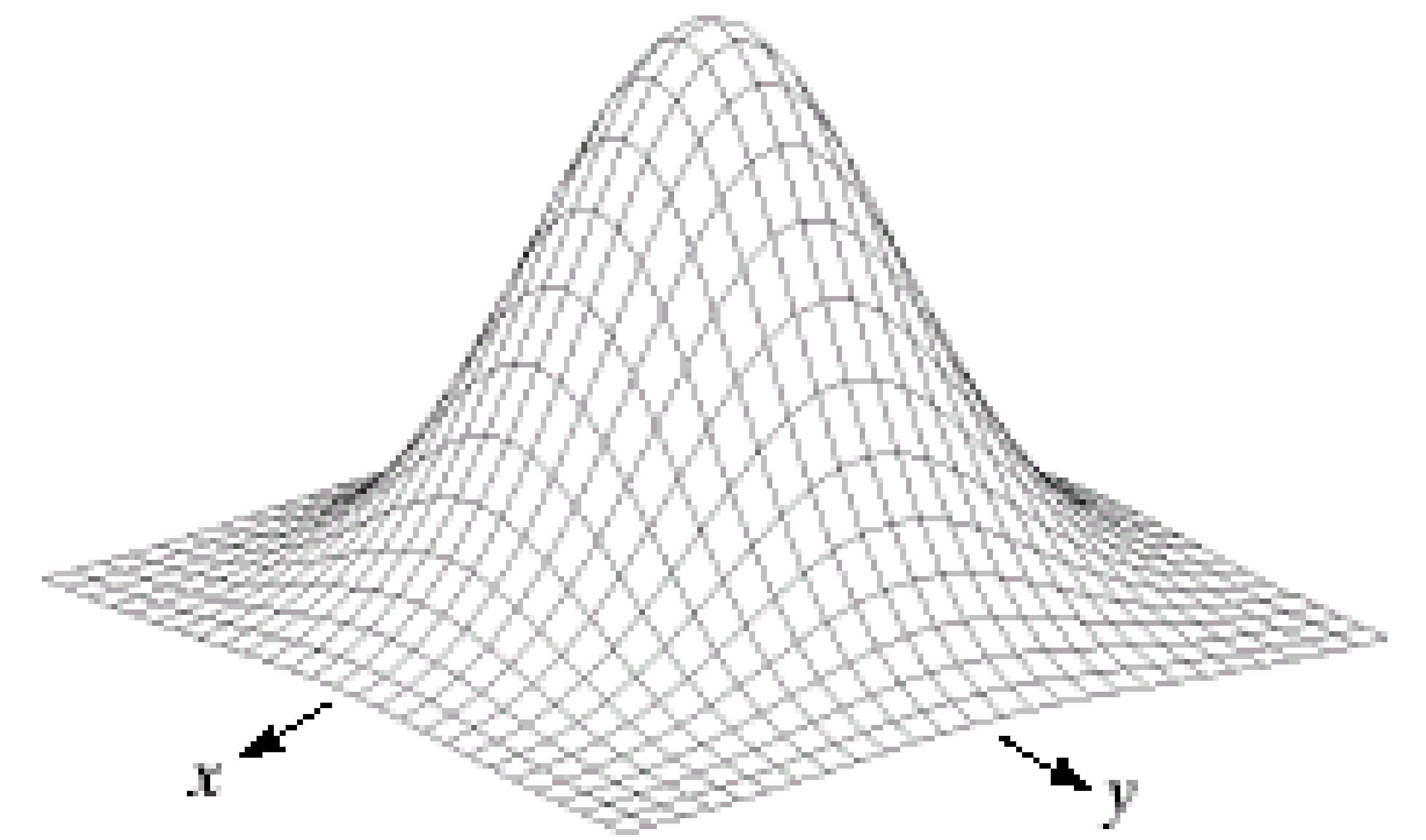
The Laplacian of h is given by –

$$\mathbf{Del^2\, h = -((r^2 - \sigma^2)/\sigma^4) \exp(-(x^2 + y^2)/\sigma^2)}$$

Laplacian

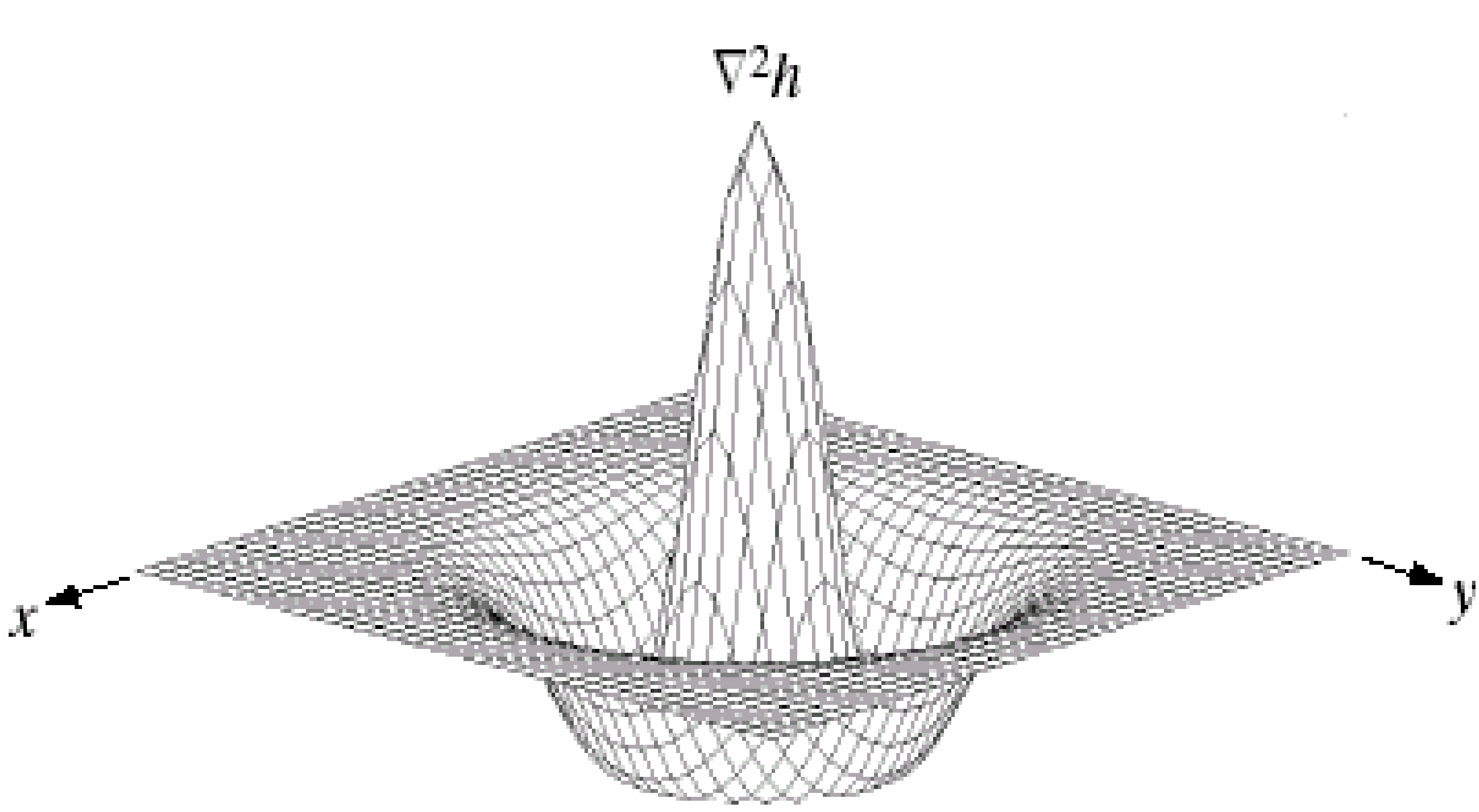
- This function is commonly called the **'Laplacian of Gaussian' (LOG)** function.
- The 3-D plot, image, cross-section and a 5×5 approximation are as shown -

Laplacian



Spatial Gaussian smoothing function

Laplacian



3-D plot

Laplacian

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

5×5 Mask

LOG

- Due to its shape, sometimes LOG is called a **'Mexican Hat'** function.
- Since the Second derivative is a linear operation, convolving the image with Del ^{2} h is the same as convolving the image with the **Gaussian Smoothing function first and then computing the Laplacian of the result.**

Disadvantages of LOG Operator

- The LOG operator being a second-derivative operator, the **influence of noise is considerable.**
- It always generates **Closed contours, which is not realistic/true.**
- **Contours means-**the outline of a figure or body; the edge or line that defines or bounds a shape or object.

Summary

- Zero crossing is used when edges are blurred and /or when high noise content is present.
- The **zero crossings allow reliable edge location and the smoothing properties of Del ^{2} h reduce the effect of noise.**
- It has very high computational complexity and requires more time.

Canny Edge Detector

- Canny introduces several ideas to help (1984).
- Reference:-Canny J.F., "A Computational Approach to edge detection," IEEE transaction Pattern Analysis and Machine Intelligence, 8(6), 678-698, Nov 1986.

Canny Edge Detector - Objectives

- **Good Edge detection:-**Algorithm should detect only the real edge points and discard all false edge points.
- **Good edge localization-**The algorithm should have the ability to produce edge points that are closure to the real edges.
- **Only one response to the each edge:-**The algorithm should not produce any false, double or spurious edges.

Canny Edge Detector

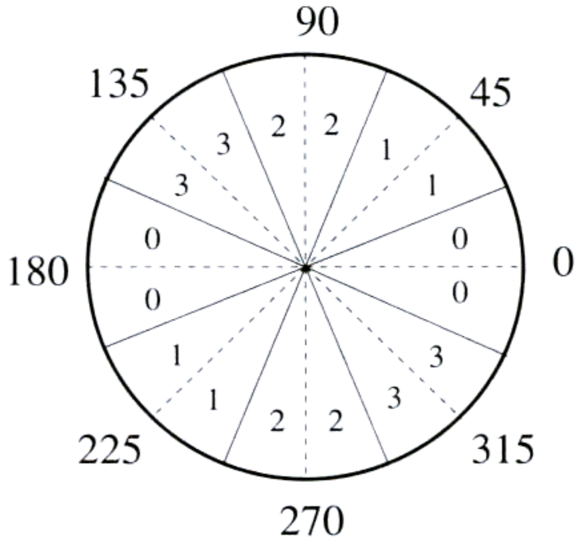
- The three phases in the Canny Edge detection algorithm are
- Smoothing and Differentiation
- Non-Maximal Suppression
- Thresholding

Smoothing and Differentiation

- In this phase the input image is smoothened with a Gaussian and the Gradient is obtained in the x and y directions.
- Store the edge magnitude and edge orientation separately in two arrays, $M(i,j)$ and $\alpha(i,j)$, respectively.

Non-Maximal Suppression

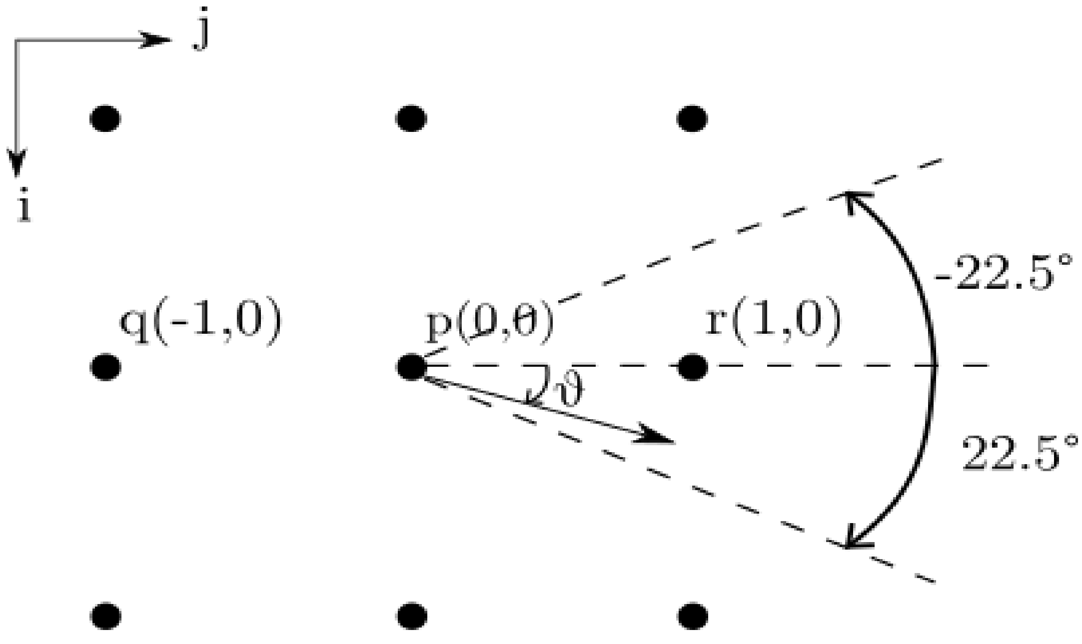
- The next step is to thin the edges.
- This is done using a process call Non maxima suppression.
- Examining every edge point is a computationally intensive task.
- To avoid such intense computations, **the gradient direction is reduced to just four sectors.**
- How the **range of 0-360^0** is divided into eight equal portions.
- Two equal portions are designated as one sector.
- Therefore, there will be four sectors.



Non-Maximal Suppression

Non maximum suppression requires us to divide the 3x3 grid of pixels into 8 sections.

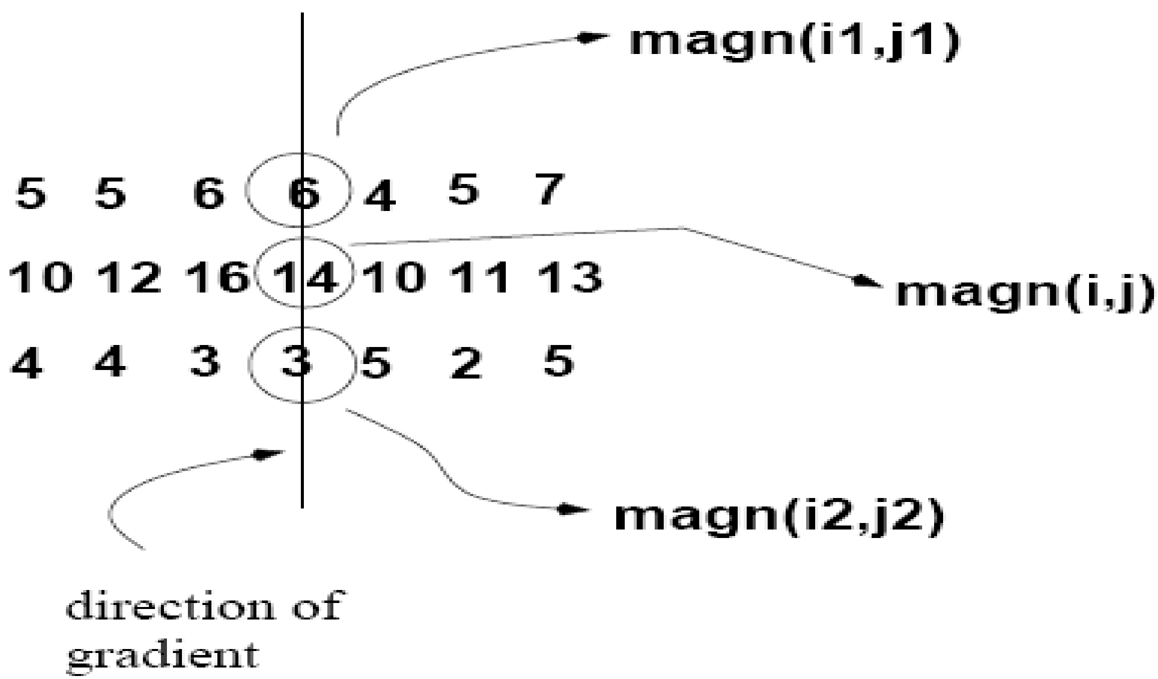
Ie. if the gradient direction falls in between the angle -22.5 and 22.5, then we use the pixels that fall between this angle (r and q) as the value to compare with pixel p, see image below.



Non-Maximal Suppression

- The gradient direction of the edge point is first approximated to one of these sectors.
- After the sector is finalized, let us assume a point of M(i,j) .
- The edge magnitude M(i1,j1) and M(i2,j2) of two neighboring pixels that falls on the same gradient direction, are considered.
- **If the magnitude of the point M(i,j) is less than the magnitude of the points M(i1,j1) or M(i2,j2), then the value is suppressed , that is the value is set to zero; otherwise the value is retained.**

Non-Maximal Suppression

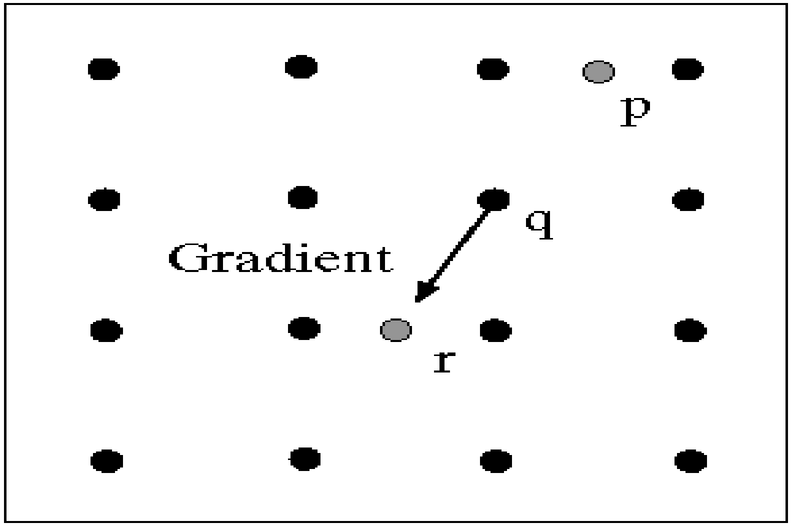


Algorithm

For each pixel (i,j) do:

if $magn(i, j) < magn(i_1, j_1)$ or $magn(i, j) < magn(i_2, j_2)$
then $I_N(i, j) = 0$
else $I_N(i, j) = magn(i, j)$

Non-Maximal Suppression



Requires checking interpolated pixels p and r

Non maximum suppression works by finding the pixel with the maximum value in an edge.

In the above image, it occurs when pixel q has an intensity that is larger than both p and r where pixels p and r are the pixels in the gradient direction of q.

If this condition is true, then we keep the pixel, otherwise we set the pixel to zero (make it a black pixel)

Thresholding

- **Reduce number of false edges by applying a threshold T**
  - All values below T are changed to 0.
  - Selecting a good values for T is difficult.
  - Some false edges will remain if T is too low.
  - Some edges will disappear if T is too high.
  - Some edges will disappear due to softening of the edge contrast by shadows.

Thresholding

- Thresholding is used to eliminate edges lower than the threshold T1, keeping edges greater than the threshold T2.
- The edges between T1 and T2 are kept if they are connected to another edge.

Double Thresholding

- It sets two thresholds, a high and a low threshold.
- In the algorithm, normalized all the values such that they will only range from 0 to 1.
- Pixels with a high value are most likely to be edges. For example, you might choose the high threshold to be 0.7, this means that all pixels with a value larger than 0.7 will be a strong edge.
- You might also choose a low threshold of 0.3, this means that all pixels less than it is not an edge and you would set it to 0.

Double Thresholding

- The values in between 0.3 and 0.7 would be weak edges.
- In other words, we do not know if these are actual edges or not edges at all.

Edge Tracking by Hysteresis

- **Hysteresis** is the dependence of the state of a system on its history.
- Now that we have determined what the strong edges and weak edges are, we need to determine which weak edges are actual edges.
- To do this, we have to perform an edge tracking algorithm. Weak edges that are connected to strong edges will be actual/real edges.
- Weak edges that are not connected to strong edges will be removed.

Canny Edge Detector – Example

Original Image

Canny Edge Detector – Example

Gradient magnitude

Canny Edge Detector – example

Thresholded gradient magnitude

Canny Edge Detector – Example

Thinning (non-maxima suppression)

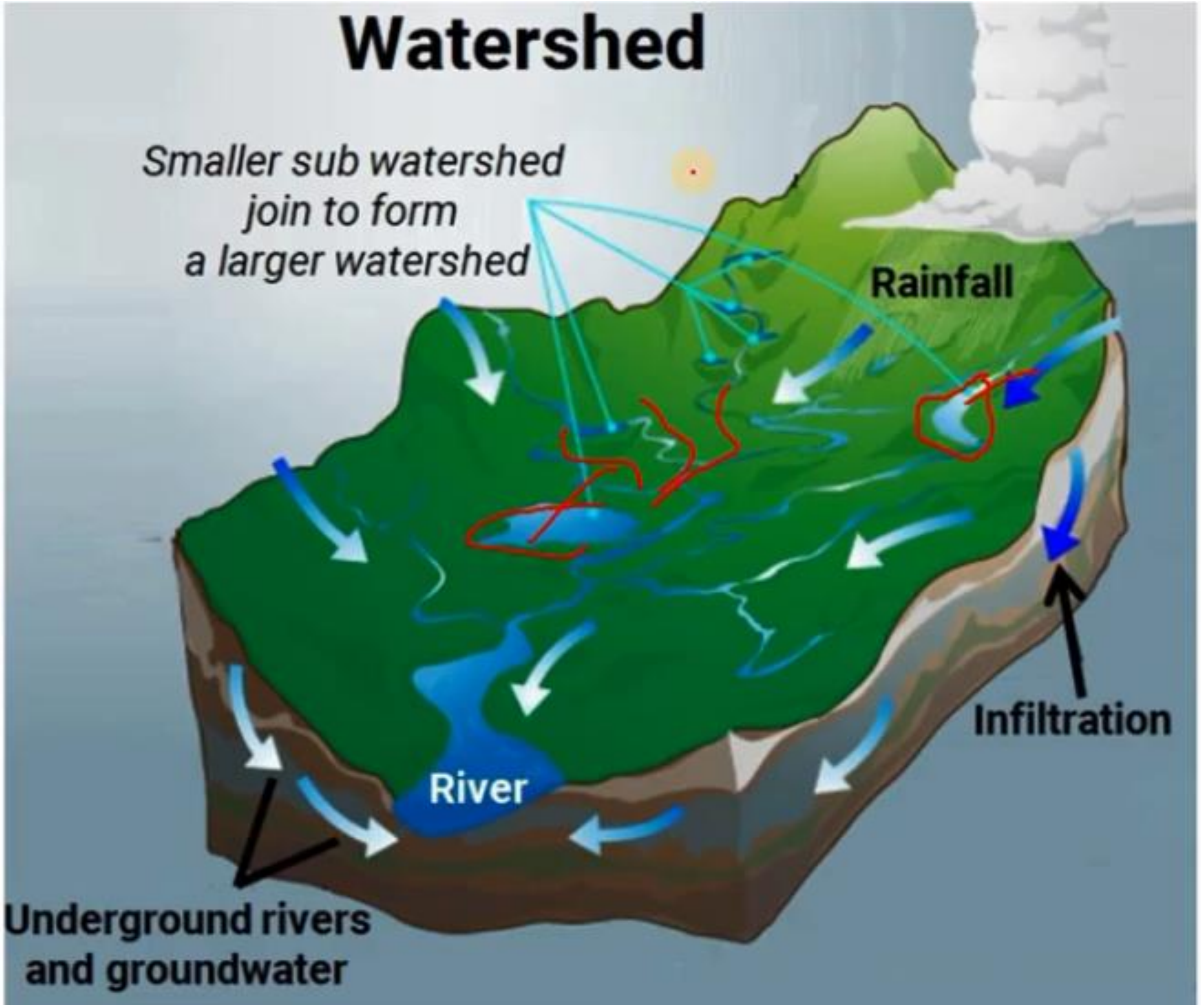
Comparison- Canny & LOG Edge Detector

Canny Edge Detector	LOG Edge Detector
1. It is not Isotropic	1. It is Isotropic (Invariant to rotation)
2.It uses First derivative to compute edge.	2. It uses second derivative to compute edge.
3. It is more likely to produce Long, thin contours because of non-maximum suppression , and thresholding.	3. This is not so in LOG

Watershed Segmentation

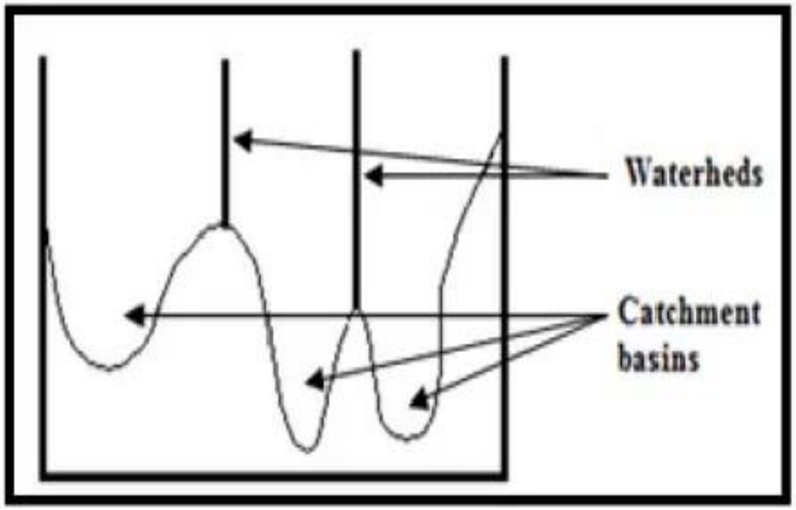
- An early powerful tool for **Image Segmentation (1979).**
- **Watershed** meaning-an area or ridge of land that separates waters flowing to different rivers, basins, or seas.
- In a Watershed transformation, the image is considered as **a 3D topographic surface.**
- The gray level of the image represents the **Altitudes.**
- The region with the constant gray level constitutes the flat zone of an image.
- Also, region **edges** correspond to **high watersheds** and **low-gradient region** interiors corresponds to **Catchment Basins.**
- **Watershed** are defined as the line separating catchment basins, which belongs to different minima.
- Segments an image into several **“Catchment Basins"** or **“Regions“**

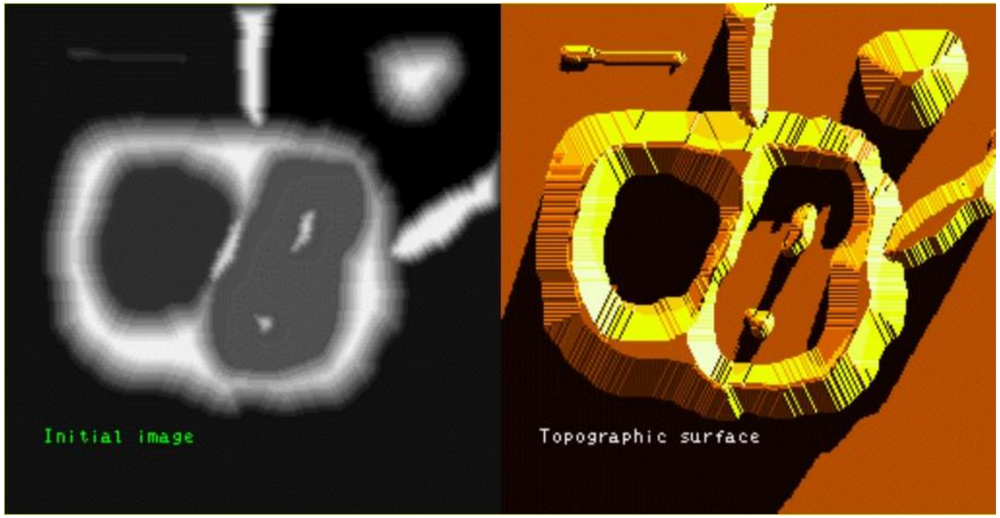
Watershed Segmentation



Watershed Segmentation

- **A watershed** can be images as a high mountain that separates two regions.
- The regions that the watershed separates are called **Catchment Basins.**

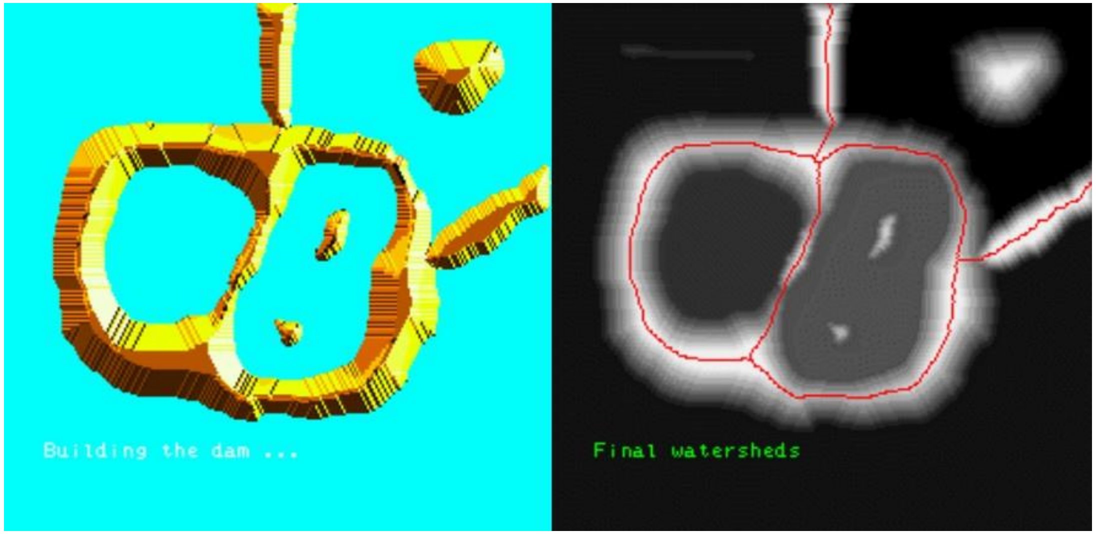




- Image can be segmented into regions where rainwater would flow into the same lake

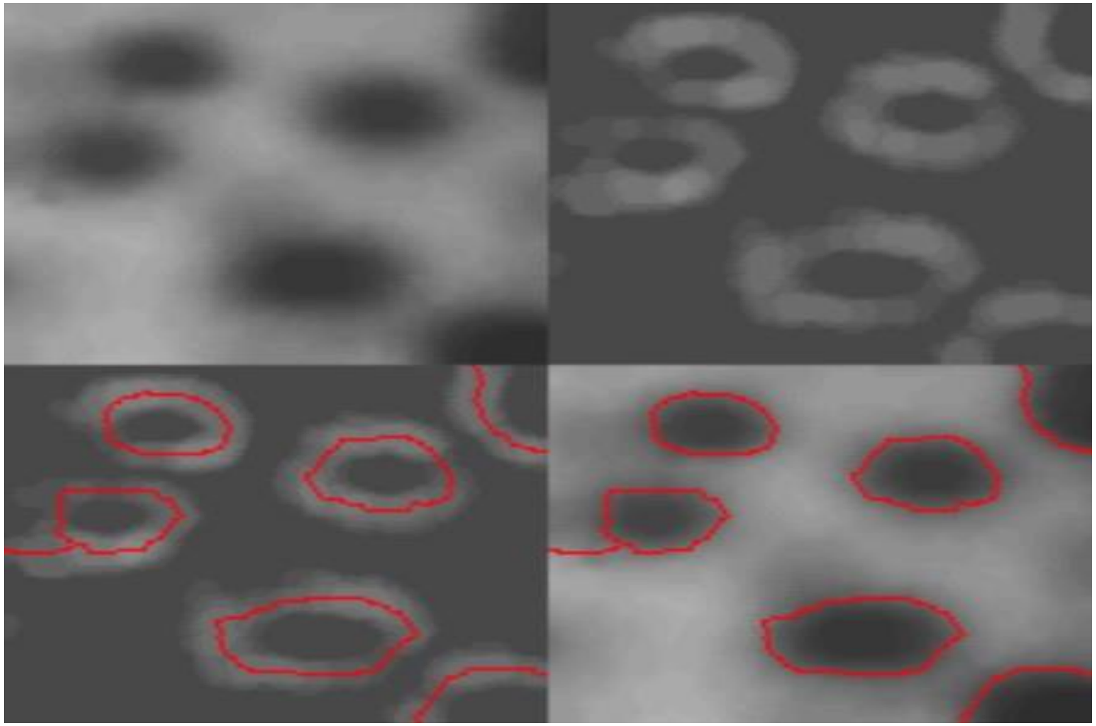
Watershed Segmentation

- Flood the landscape from local minima and prevent merging of water from different minima
- Results in partitioning the image into **Catchment Basins** and **Watershed Lines**



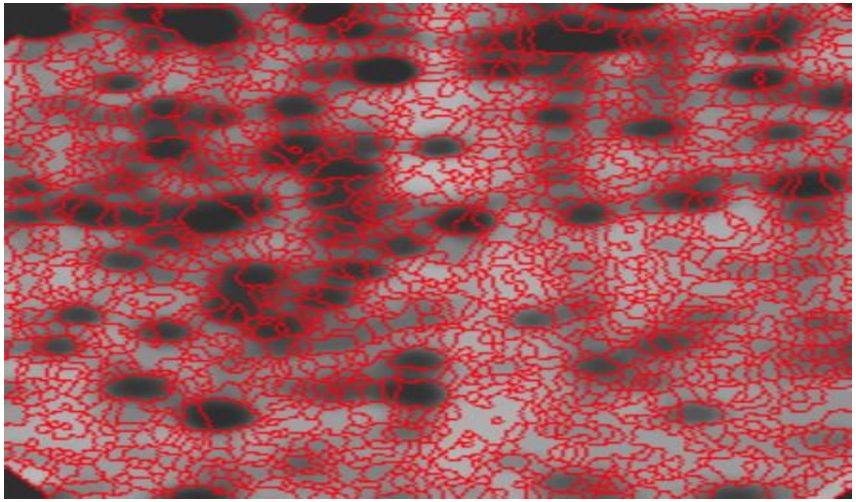
Watershed Segmentation

- ❖ Generally applied on **Image Gradients** instead of applying directly on images.
- ❖ (Top left) Original image; (Top right) Gradient image; (Bottom left) Watersheds of gradient image; (Bottom right) Final segmentation output



Watershed Segmentation Drawbacks

- **Over Segmentation:-**When the watershed transform infers Catchment Basin from the gradient of the image. It contains a **Myriad** of **Small regions**.



- **Soln- Marker based Watershed Transformation:-**
- Two markers are defined **a. Foreground marker,** and **b. Background marker**
- Gradient image is modified to keep only the most significant contours in the areas of interest between the markers.

Watershed Segmentation Drawbacks

- **Sensitivity to Noise:-**The noise in the image results in Over Segmentation. Filters are used to remove noise.
- **Low Contrast Boundaries:-**If the signal to noise is not high enough at the contour of interest, the Watershed transform will be unable to detect contours with higher value between markers.
- **Poor Detection of Thin Edges:-**Gradient image make it difficult to detect thin edges.

Watershed Segmentation-Example

- Read a image and perform the watershed transformation. Now corrupt the image by adding the salt and pepper noise and perform the watershed transformation for this Noisy image. Comment on the result.

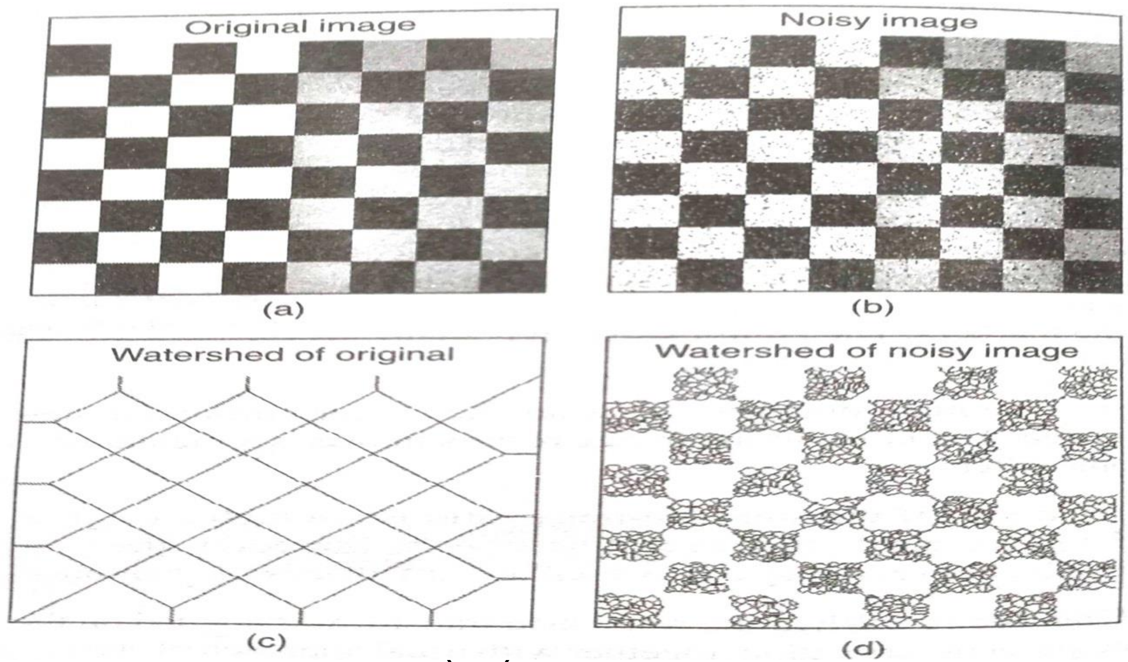
Soln-a=checkerboard(32);

a1=imnoise(a, salt & pepper ,0.1);

b=watershed(a,4);

b1=watershed(a1,4);

imshow(a), fig., imshow(a1), fig., imshow(b), fig., imshow(b1)



Region Growing

- Region growing is the a process of grouping the pixels or sub regions to get a bigger region present in an image.
- Important issues of the region growing algorithm are the selection of the initial seed, seed growing criteria, and termination of the segmentation process.
- User has to find out the seed that represents the ROI.
- The seeds can be either single or multiple.

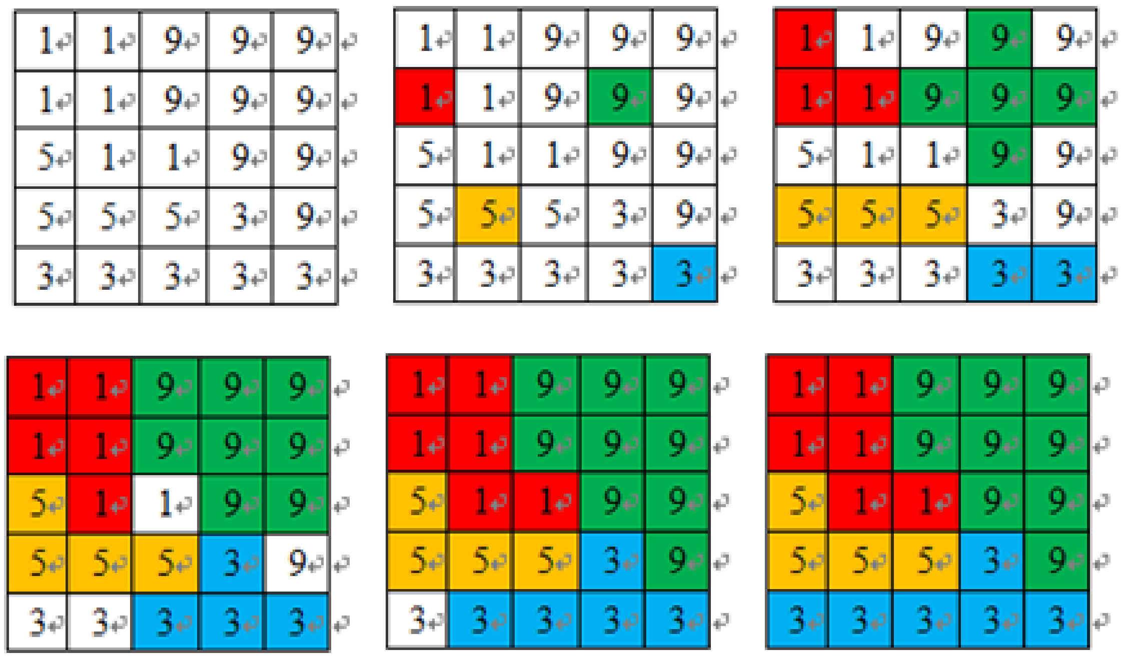
Region Growing

❖ Algorithm:

a. Choose a random pixels-Key/Seed pixels

b. Use 8-connected and threshold to determine

c. Repeat a and b until almost points are classified.



Region Growing

Example 9.2 Consider the image shown in Fig. 9.27. Show the results of the region-growing algorithm.

Solution Assume that the seed points are indicated at the coordinates (2, 4) and (4, 2). The seed pixels are 9 and 1 (underlined in Fig. 9.27). Let them be $s1$ and $s2$. Let the seed pixels 1 and 9 represent the regions C and D, respectively.

Subtract the pixel value from the seed value that represents the region. If the difference between the pixel value and seed point is 4 (i.e., $T = 4$), merge the pixel with that region. Otherwise, merge the pixel with the other region. The resultant image is shown in Fig. 9.28.

1	0	7	8	7
0	1	8	<u>9</u>	8
0	0	7	9	8
0	<u>1</u>	8	8	9
1	2	8	8	9

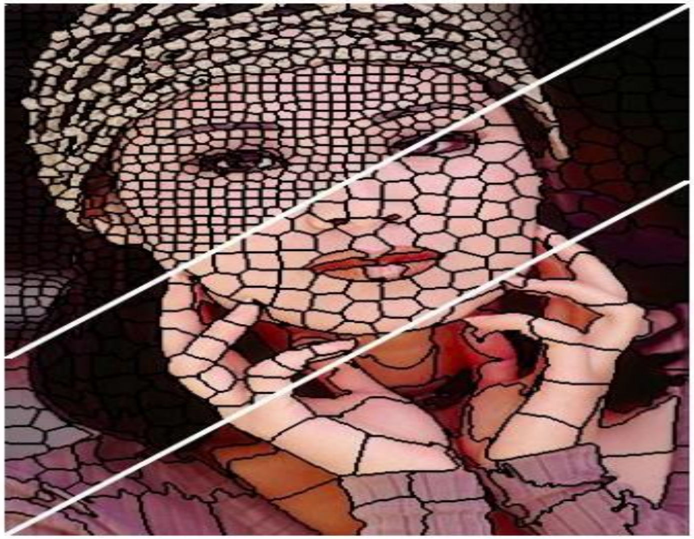
Fig. 9.27 Region-growing algorithm—original image

C	*C*	*D*	*D*	*D*
C	*C*	*D*	*D*	*D*
C	*C*	*D*	*D*	*D*
C	*C*	*D*	*D*	*D*
C	*C*	*D*	*D*	*D*

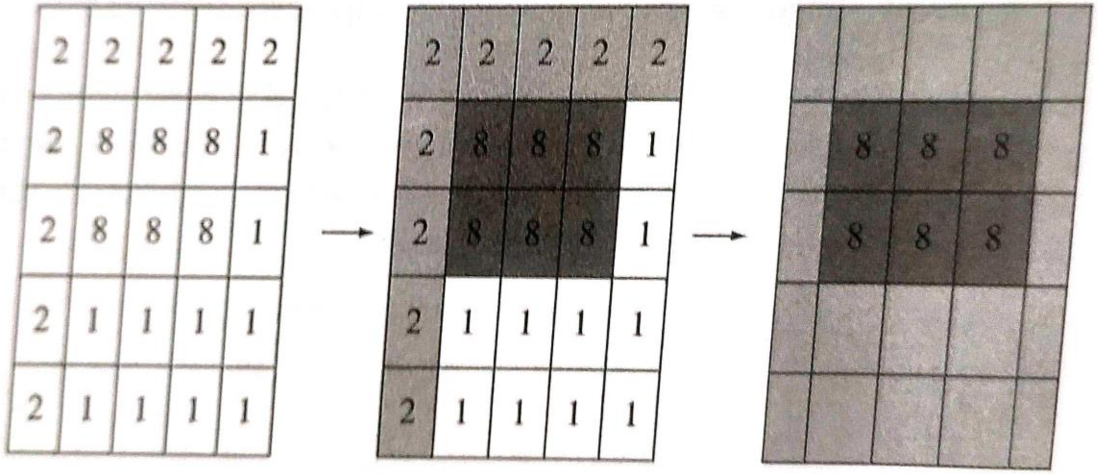
Fig. 9.28 Region-growing algorithm—resultant image for Example 9.2

Region Splitting and Merging

- Region splitting methods involve splitting the image into successfully finer regions.
- Region merging methods successively merge pixels into groups based on various heuristics such as color differences
- Figure shown below an image segmented into such superpixels
- Generally used as preprocessing step to higher-level segmentation algorithms.



Region Splitting and Merging



(a) (b) (c)

Fig. 9.31 Split-and-merge technique (a) Original image (b) Image showing separate regions (c) Final result

Split & Merge-Quad tree

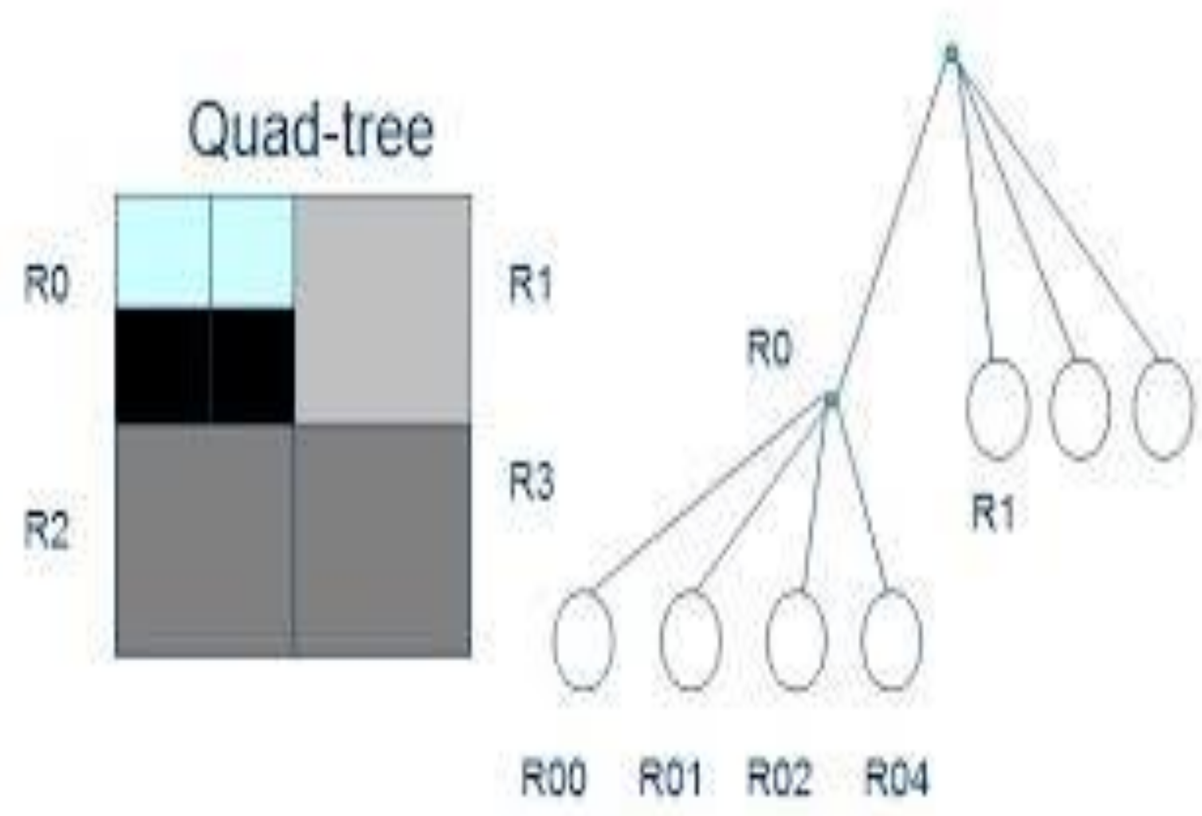


Image Segmentation

- ❖ Usually **Image Segmentation** is an initial and vital step in a series of processes aimed at **overall image understanding.**
- ❖ **Applications of Image Segmentation include-**
- ➢ Identifying objects in a scene for object-based measurements such as size and shape
- ➢ Identifying objects in a moving scene for object-based video compression (MPEG4)
- ➢ Identifying objects which are at different distances from a sensor using depth measurements from a laser range finder enabling path planning for a mobile robots
- ➢ To detect object


Diolch
Kiitos
Sheun
umesc
Dekuju/Dekujeme
Kasih
Todah
Mamnoon
Shnorhakalutiun
Te
ekkür
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ači
Gamsahapnida
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
or
Dhanyavaad
al
Dakujem
Waad
Dhanyavaadalu
Kop
Salamat
Merci
Dankie
Go
Grazie
Faleminderit
Daw
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun
Hain
Dhan
daa

Smoothing & Sharpening Techniques in Image Processing

By,

Prof (Dr.) P P Ghadekar

Recapitulation

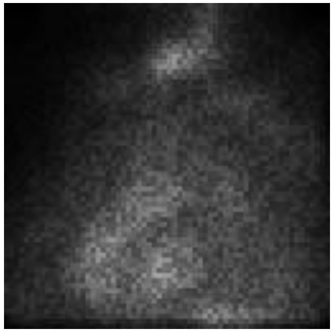
- Point Processing Techniques
  - Image Negatives
  - Logarithmic transformation for Dynamic range Compression.
  - Power Law transformation
  - Contrast Stretching
  - Gray Level Slicing
- Histogram Based Processing Techniques
  - Histogram Equalization
- Histogram Modification/SpecificationImage Differencing
- Image Averaging

Content

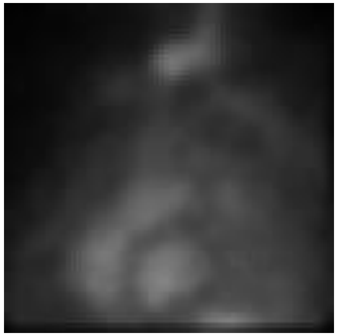
- On completion Student will be able to understand
- Introduction
- Mask Processing Techniques
- Brief review of linear operators
- Linear image smoothing techniques
- Nonlinear/Median image smoothing techniques

Introduction

Why do we need image smoothing?



What is “image” and what is “noise”?



- Frequency spectrum
- Statistical properties

Local or Neighborhood Operation

- In neighborhood operation, the pixels in an image are modified based on some function of the pixels in their neighborhood.
- Linear spatial filtering is often referred to as convolving a mask with an image.
- The filter masks are sometimes called convolution masks or convolution kernels.
- Spatial Filtering
  - Spatial filtering involves passing a weighted mask or kernel over the image and replacing the original image pixel value corresponding to the Center of the kernel with the sum of the original pixel values in the region corresponding to the kernel multiplied by the kernel weights.

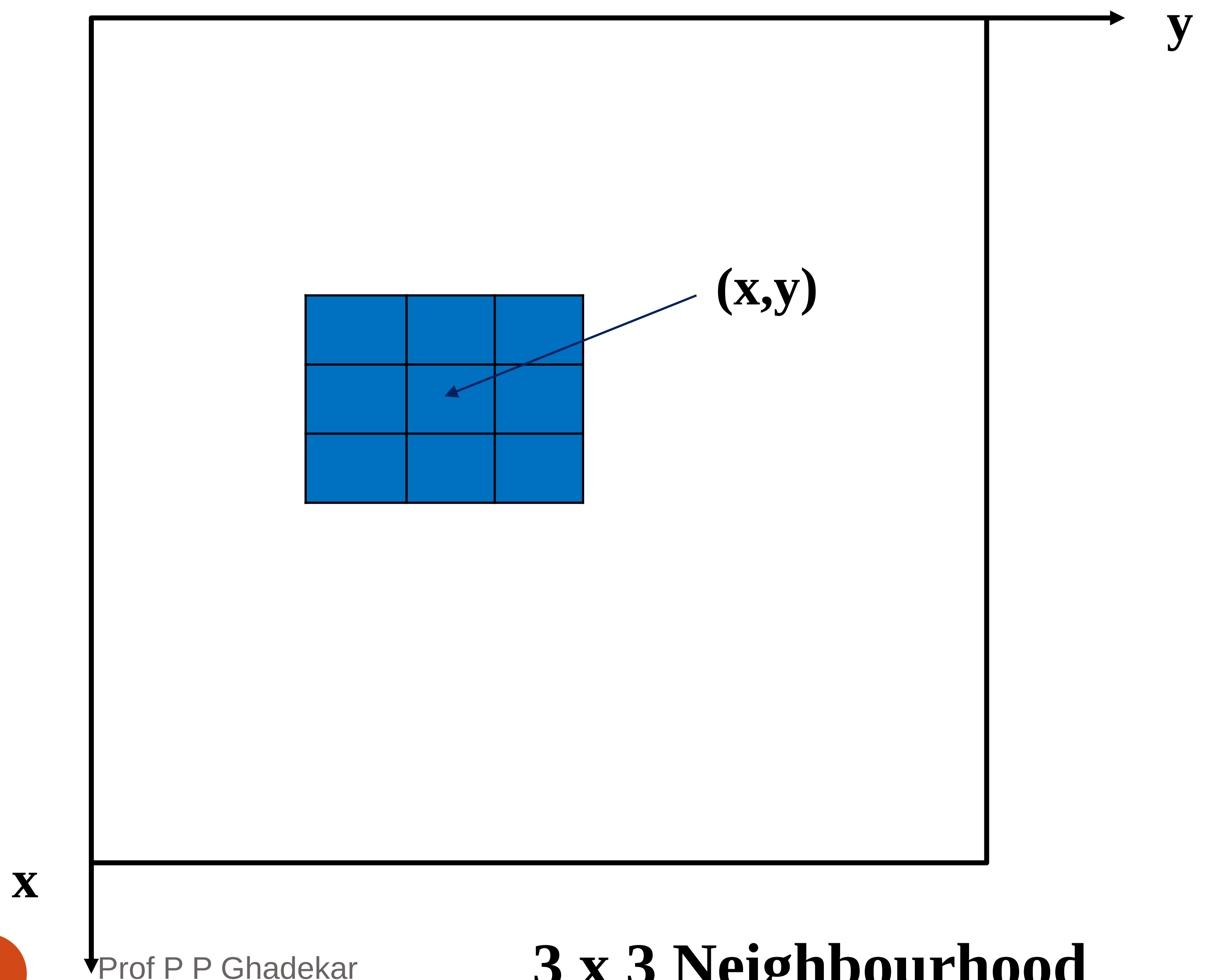
Linear Filtering

- In linear filtering , each pixel in the input image is replaced by a linear combination of intensities of neighboring pixels.
- That is each pixel value in the output image is a weighted sum of the pixels in the neighborhood of the corresponding pixel in the input image.
- Linear filtering can be used to smoothen an image as well as sharpen the image.
- A spatially invariant linear filter can be implemented using a convolution mask.
- If different filter weights are used for different parts of the image then the linear filter is a spatially varying.

Mean Filter or Average Filter or Low pass Filter

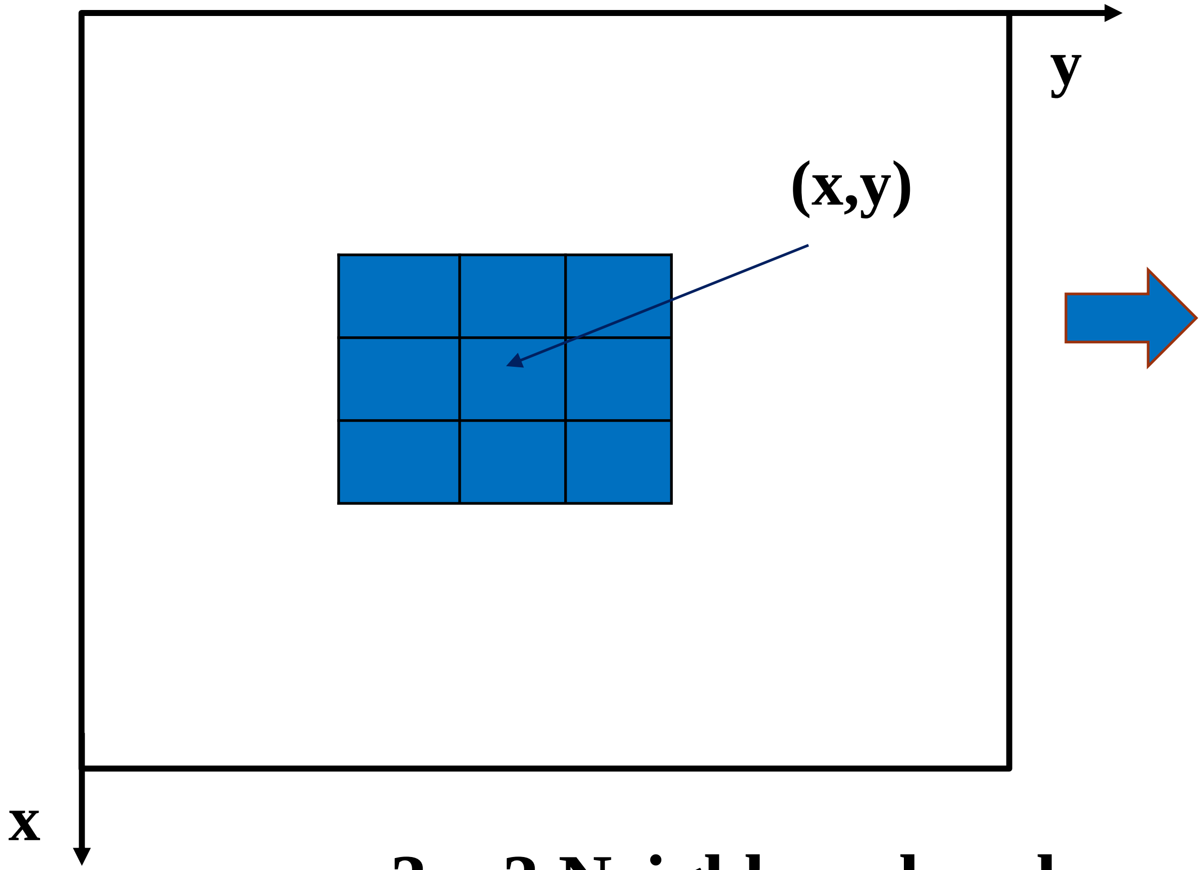
- The mean filter replaces each pixel by the average of all the values in the neighborhood. The size of the neighborhood controls the amount of filtering.
- In a spatial averaging operation, each pixel is replaced by a weighted average of its neighborhood pixels.
- The low pass filter preserves the smooth region in the image and removes the sharp variations leading to blurring effect.

Neighbourhood



3 x 3 Neighbourhood

Mask Processing



3 x 3 Neighbourhood

$W_{-1,-1}$	$W_{-1,0}$	$W_{-1,1}$
$W_{0,-1}$	$W_{0,0}$	$W_{0,1}$
$W_{1,-1}$	$W_{1,0}$	$W_{1,1}$

3 x 3 Mask

$$g(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} W_{i,j}\, f(x+i, y+j)$$

Mask/Kernel

- **What is a Mask?**
- A mask is a small matrix whose values are called weights.
- The origins of symmetric masks are usually their center pixel position.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolution

- Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel.

(1) For each pixel in the input image, the mask is conceptually placed on top of the image with its origin lying on that pixel.

(2) The values of each input image pixel under the mask are multiplied by the values of the corresponding mask weights.

(3) The results are summed together to yield a single output value that is placed in the output image at the location of the pixel being processed on the input.

Smoothing Spatial Filter

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Smoothing Spatial Filter

Averaging Filter (low pass Filter)

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

3 x 3 Averaging Mask

→ Box Filter

$$\mathbf{g(x,y)} = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} \mathbf{f(x+i, y+j)}$$

Illustration of Spatial Filtering-Example

7	9	11
10	50	8
9	5	6

Original Image

1/9

1	1	1
1	1	1
1	1	1

3 x 3 Averaging Mask

0	0	0	0	0
0	7	9	11	0
0	10	50	8	0
0	9	5	6	0
0	0	0	0	0

Input Image after zero padding

Illustration of Spatial Filtering-Example

1/9 0	1/9 0	1/9 0	0	0
1/9 0	1/9 7	1/9 9	11	0
1/9 0	1/9 10	1/9 50	8	0
0	9	5	6	0
0	0	0	0	0

$$0 \times 1/9 + 0 \times 1/9 + 0 \times 1/9 + 0 \times 1/9 + 7 \times 1/9 + 9 \times 1/9 + 0 \times 1/9 + 10 \times 1/9 + 50 \times 1/9 = 8.4$$

Illustration of Spatial Filtering-Example

0	1/9 0	1/9 0	1/9 0	0
0	1/9 8.4	1/9 9	1/9 11	0
0	1/9 10	1/9 50	1/9 8	0
0	9	5	6	0
0	0	0	0	0

0*1/9+0*1/9+0*1/9+7*1/9+9*1/9+11*1/9+10*1/9+50*1/9+8*1/9=10.5

Illustration of Spatial Filtering-Example

0	0	1/9 0	1/9 0	1/9 0
0	8.4	1/9 10.5	1/9 11	1/9 0
0	10	1/9 50	1/9 8	1/9 0
0	9	5	6	0
0	0	0	0	0

0*1/9+0*1/9+0*1/9+9*1/9+11*1/9+0*1/9+50*1/9+8*1/9+0*1/9=8.66

Illustration of Spatial Filtering-Example

0	0	0	0	0
1/9 0	1/9 8.4	1/9 10.7	8.6	0
1/9 0	1/9 10	1/9 50	8	0
1/9 0	1/9 9	1/9 5	6	0
0	0	0	0	0

0*1/9+7*1/9+9*1/9+0*1/9+10*1/9+50*1/9+0*1/9+9*1/9+5*1/9=10.0

Illustration of Spatial Filtering-Example

0	0	0	0	0
0	8.4444	10.5556	8.6667	0
0	10.0000	12.7778	9.8889	0
0	8.2222	9.7778	7.6667	0
0	0	0	0	0

Result of Averaging Filter

7	9	11
10	50	8
9	5	6

Original Image

8.4444	10.5556	8.6667
10.0000	12.7778	9.8889
8.2222	9.7778	7.6667

Image after Spatial Averaging

Spatial Low Pass Filter-Example

Original Image

3 x 3
Smoothing
filter

Smoothened Image

Spatial Low Pass Filter-Example

Original Image

5 x 5

Smoothing filter

Smoothened Image

Spatial Low Pass Filter-Example

FIGURE (a) Original image, of size 500*500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes n=3, 5, 9, 15, and 35, respectively

Linear Image smoothing techniques

Box filters. Arithmetic mean L×L operator

❖ An example

$$\begin{matrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 8 & 5 & 8 & 8 & 5 & 8 & \cdots \\ \cdots & 8 & 5 & 8 & 8 & 5 & 8 & \cdots \\ \cdots & 8 & 5 & 8 & 8 & 5 & 8 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix} * \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{matrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 7 & 7 & 7 & 7 & 7 & 7 & \cdots \\ \cdots & 7 & 7 & 7 & 7 & 7 & 7 & \cdots \\ \cdots & 7 & 7 & 7 & 7 & 7 & 7 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix}$$

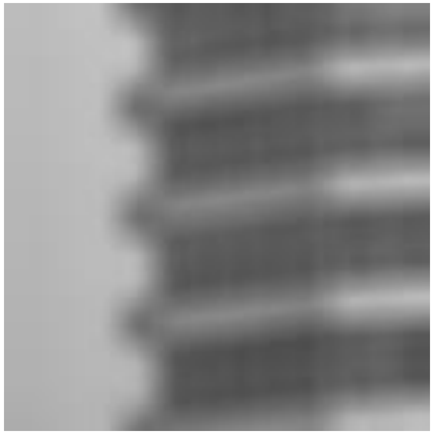
$$\begin{matrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 8 & 5 & 8 & 5 & 8 & 5 & \cdots \\ \cdots & 8 & 5 & 8 & 5 & 8 & 5 & \cdots \\ \cdots & 8 & 5 & 8 & 5 & 8 & 5 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix} * \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{matrix} & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & 6 & 7 & 6 & 7 & 6 & 7 & \cdots \\ \cdots & 6 & 7 & 6 & 7 & 6 & 7 & \cdots \\ \cdots & 6 & 7 & 6 & 7 & 6 & 7 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix}$$

Linear Image smoothing techniques

Box filters. Arithmetic mean L×L operator

- Image smoothed with 3×3, 5×5, 9×9 and 11 ×11 box filters

Original Image

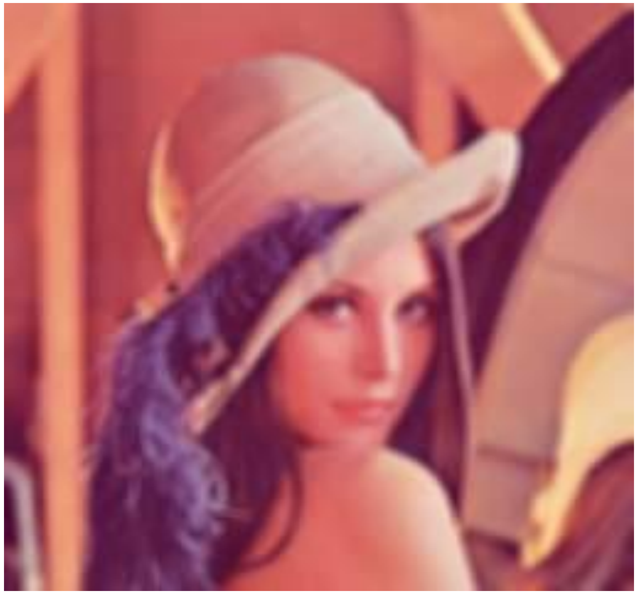


3×3 box Filter, 5×5 box Filter, 9×9 box Filter 11 ×11 box filters

Linear Image smoothing techniques

Box filters. Arithmetic mean L×L operator

Original Lena image



Lena image filtered with

5x5 box filter

Linear Image smoothing techniques

- Computes a weighted average of pixels in the window
- Less blurring, less noise cleaning for the same size
- The family of binomial filters can be defined recursively

Linear Image smoothing technique

$$\mathbf{b}^4 = \frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} * \frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix}$$

$$\mathbf{b}8 = \frac{1}{16}\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix} * \frac{1}{16}\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix} = \frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\4 & 16 & 24 & 16 & 4\\6 & 24 & 36 & 24 & 6\\4 & 16 & 24 & 16 & 4\\1 & 4 & 6 & 4 & 1\end{bmatrix}$$

Weighted Filter

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3*3 Weighted mask

Smoothing Spatial Filter

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

3*3 Weighted mask

$$g(x,y) = 1/16 \sum_{i=-1}^{1} \sum_{j=-1}^{1} W_{i,j}\, f(x+i, y+j)$$

Spatial Filter Generation Operation

$$g(x,y) = \frac{\sum_{i=-a}^{a}\sum_{j=-b}^{b} W_{ij}\, f(x+i, y+j)}{\sum_{i=-a}^{a}\sum_{j=-b}^{b} W_{ij}}$$

Mask Size = M X N

Where, M=2a+1, N=2b+1

Average Vs. Weighted Average

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

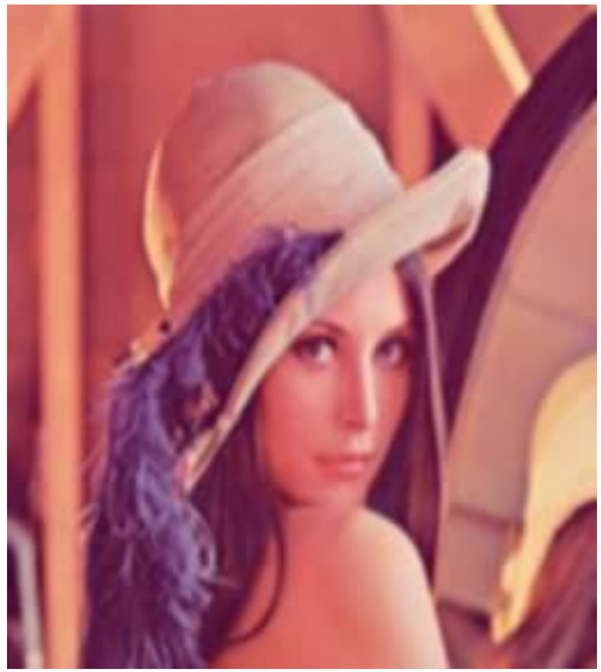
Original Image

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

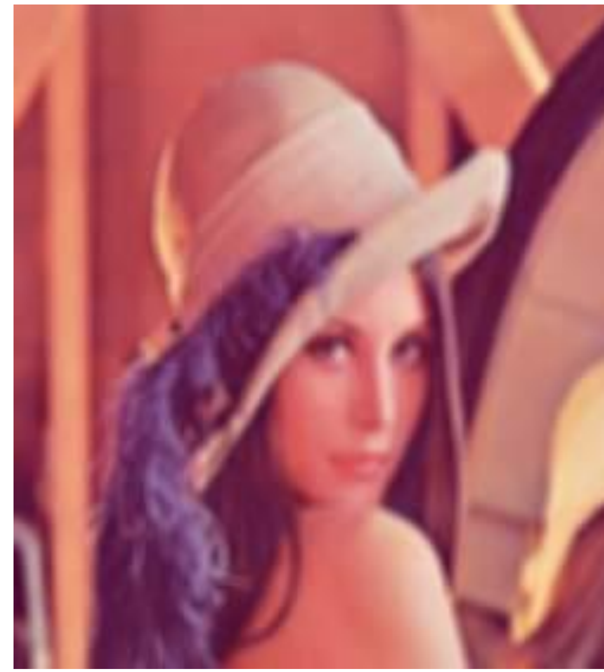
Linear Image smoothing techniques

Weighted filters. Example

Original Lena image



Lena image filtered with Weighted 5x5 kernel

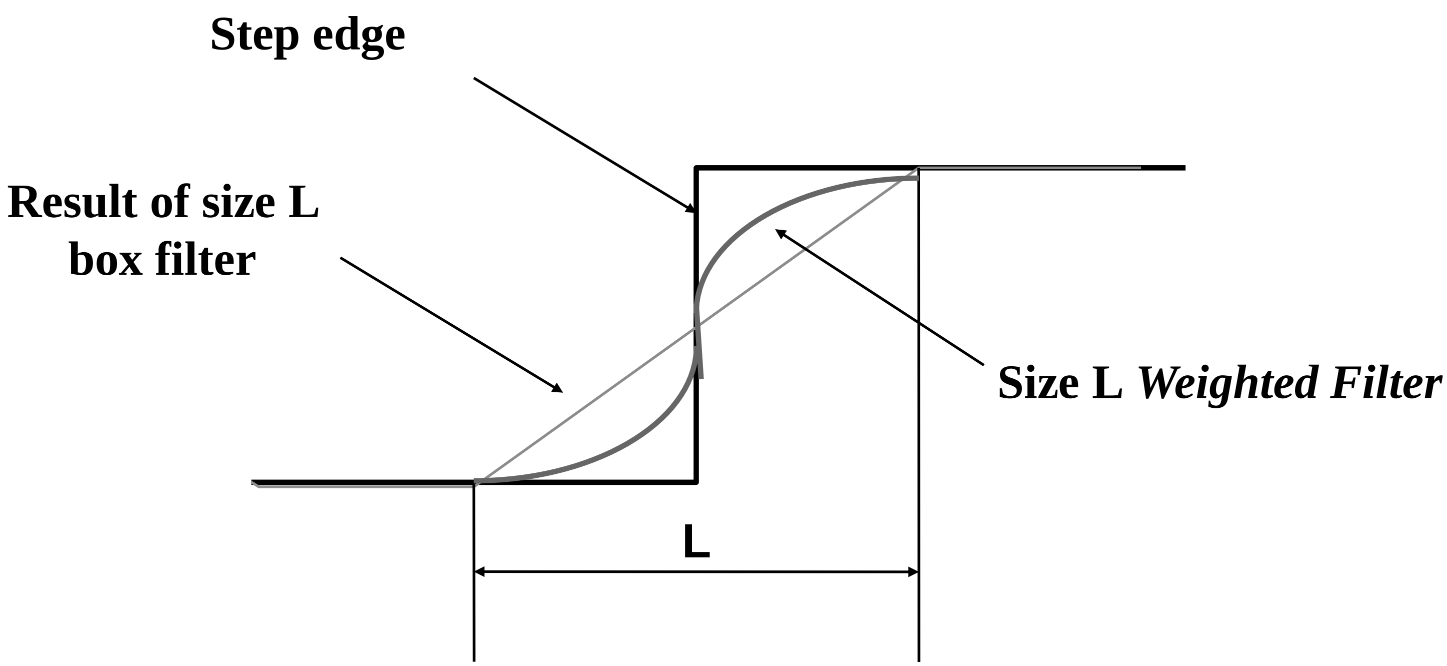


Lena image filtered with box filter 5x5

Linear Image smoothing techniques

Weighted and box filters. Edge blurring comparison

- Linear filters have to compromise smoothing with edge blurring



Order-Statistics/Median Filters

- Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- The best-known example in this category is the ***median filter***, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median).
- Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.
- Median filters are particularly effective in the presence of ***impulse noise***, also called ***salt-and-pepper noise*** because of its appearance as white and black dots superimposed on an image.

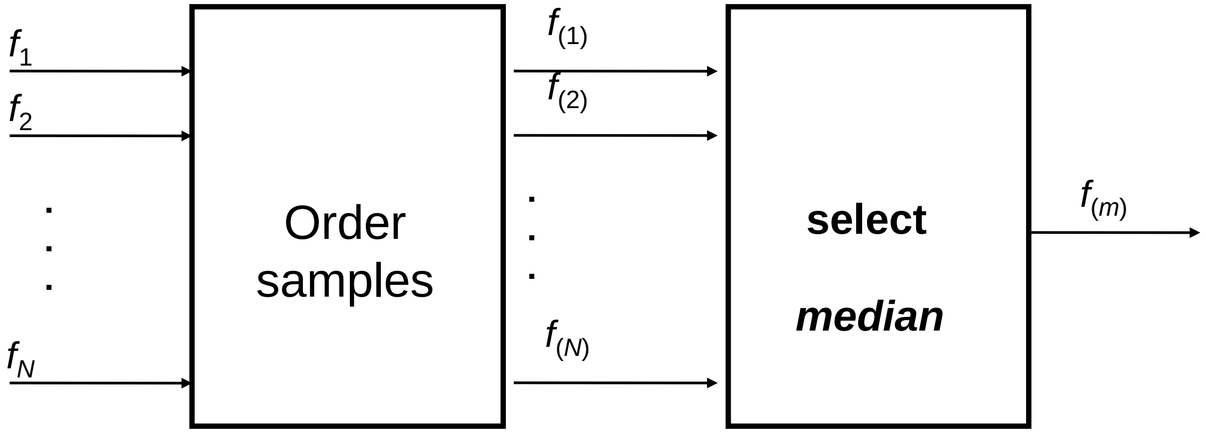
Median Filter

- The median of a set of values is such that half the values in the set are less than or equal to median value, and half are greater than or equal to median value.
- In order to perform median filtering at a point in an image, we first sort the values of the pixel in question and its neighbors, determine their median, and assign this value to that pixel.
- For example, in a 3*3 neighborhood the median is the 5th largest value, in a 5*5 neighborhood the 13th largest value, and so on.
- When several values in a neighborhood are the same, all equal values are grouped. For example, suppose that a 3*3 neighborhood has values(10, 20, 20, 20, 15, 20, 20, 25, 100).
- These values are sorted as (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in a median of 20.

Nonlinear image smoothing

The median filter [Pratt 1991]

- Block diagram

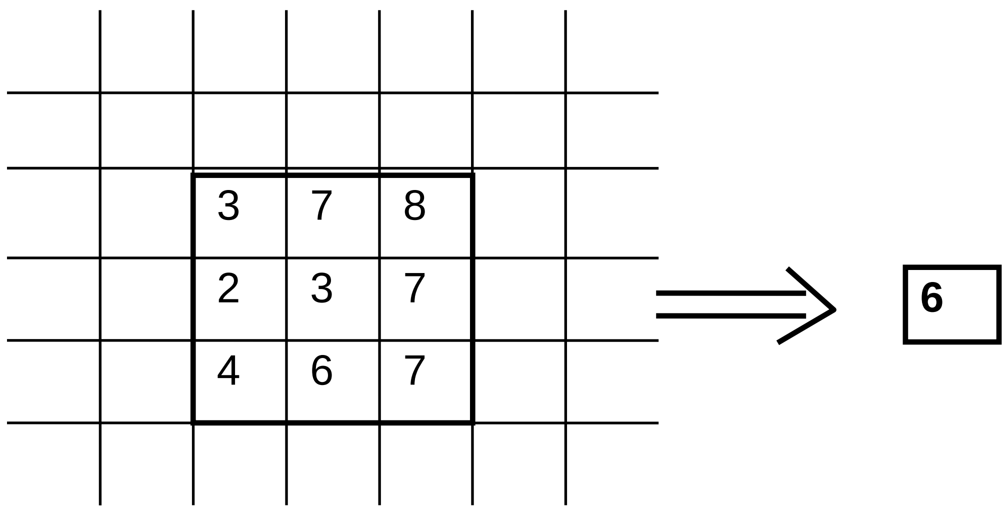


$$m = \frac{N+1}{2} \qquad N \text{ is odd}$$

Nonlinear image smoothing

The median filter

- Numerical example



2, 3, 3, 4, **6**, 7, 7, 7, 8

Nonlinear image smoothing

The median filter

- It support Nonlinearity property.
- $\text{median}\{ \boldsymbol{f}1 + \boldsymbol{f}2\} \neq \text{median}\{ \boldsymbol{f}1\} + \text{median}\{ \boldsymbol{f}2\}$.

However:

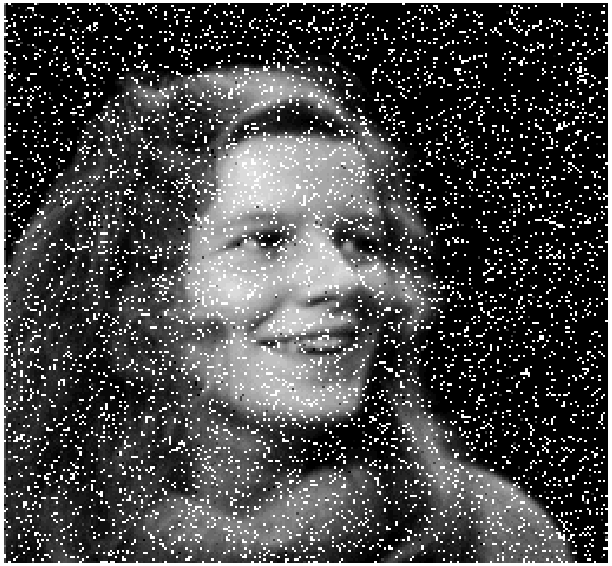
$\text{median}\{ c\,\boldsymbol{f} \} = c\ \text{median}\{ \boldsymbol{f} \}$,

$\text{median}\{ c + \boldsymbol{f} \} = c + \text{median}\{ \boldsymbol{f} \}$.

- The filter selects a sample from the window, does not average
- Edges are better preserved than with linear filters.
- Best suited for “salt and pepper” (Impulse noise) noise.

Nonlinear image smoothing

The median filter



Noisy image

5x5 median filtered

5x5 box filter

Nonlinear image smoothing
The median filter

- Implementing the median filter
- Sorting needs $O(N^2)$ comparisons
  - Bubble sort
  - Quick sort
  - Huang algorithm (based on histogram)

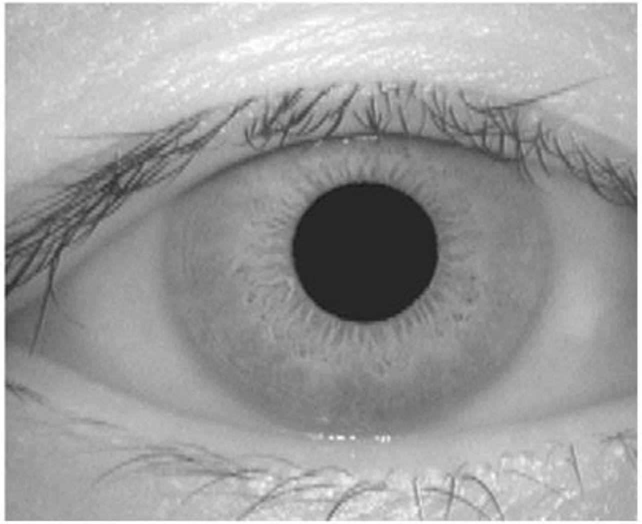
- **Caution: points, thin lines and corners are erased by the median filter**

Nonlinear image smoothing

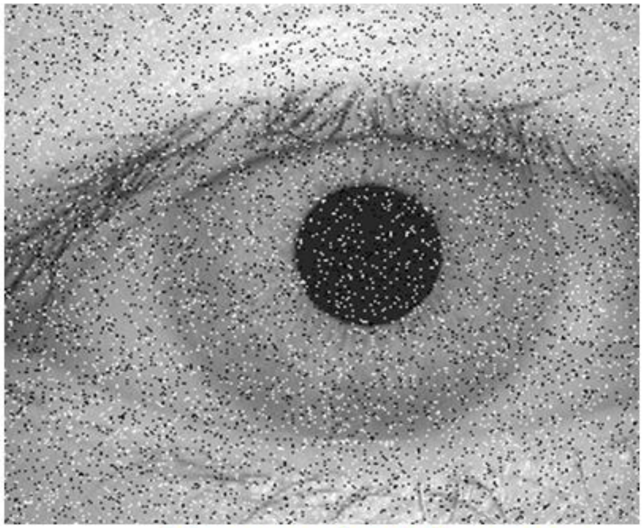
The median filter

- ➢ Example of color median filtering
- ➢ 5x5 pixels window
- Image (a) original image
- Image (b) filtered image

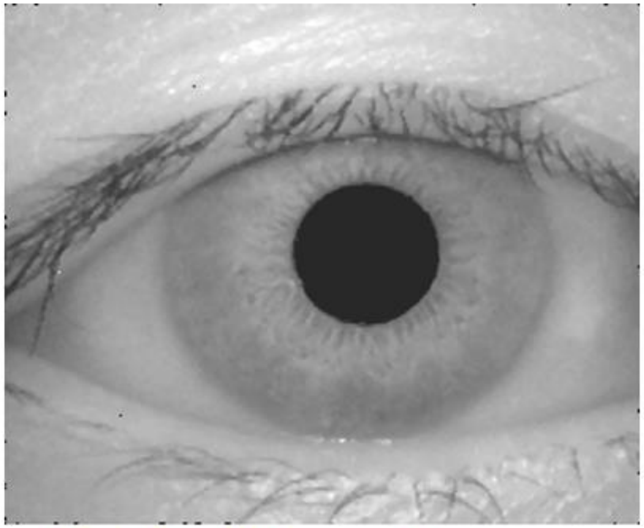
Median filter



Original Image



Corrupted Image



Median filtered Image

Drawback of Median Filter

- Removes Image Detail
- Signal Dependent Noise

Nonlinear image smoothing

The weighted median filter

- The basic idea is to give higher weight to some samples, according to their position with respect to the center of the window.
- Each sample is given a weight according to its spatial position in the window.
- Weights are defined by a weighting mask.
- Weighted samples are ordered as usually.
- The weighted median is the sample in the ordered array such that neither all smaller samples nor all higher samples can cumulate more than 50% of weights.
- If weights are integers, they specify how many times a sample is replicated in the ordered array.

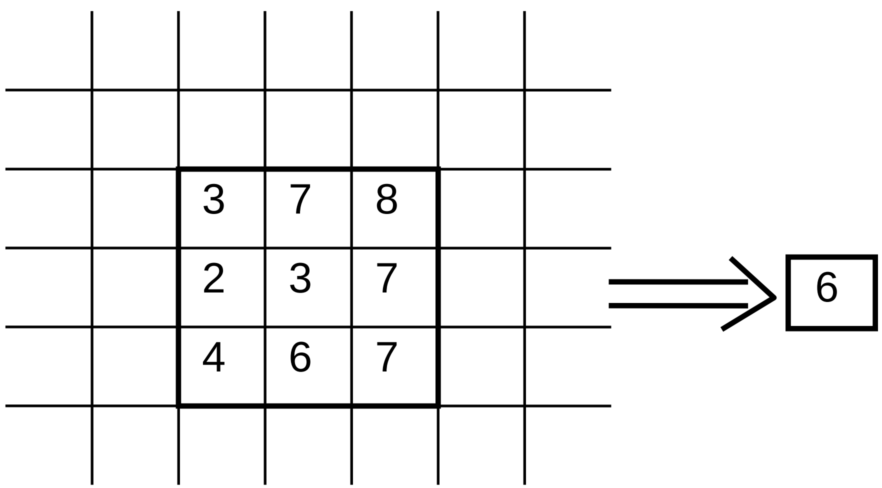
Nonlinear image smoothing

The weighted median filter

- Numerical example for the weighted median filter

1	2	1
2	3	2
1	2	1

Mask Filter Specify how many times a sample is replicated in the ordered array



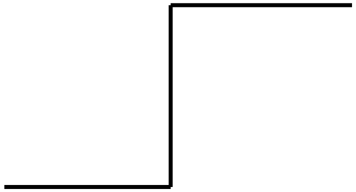
2, 2, 3, 3, 3, 3, 4, **6**, 6, 7, 7, 7, 7, 7, 8

Edges

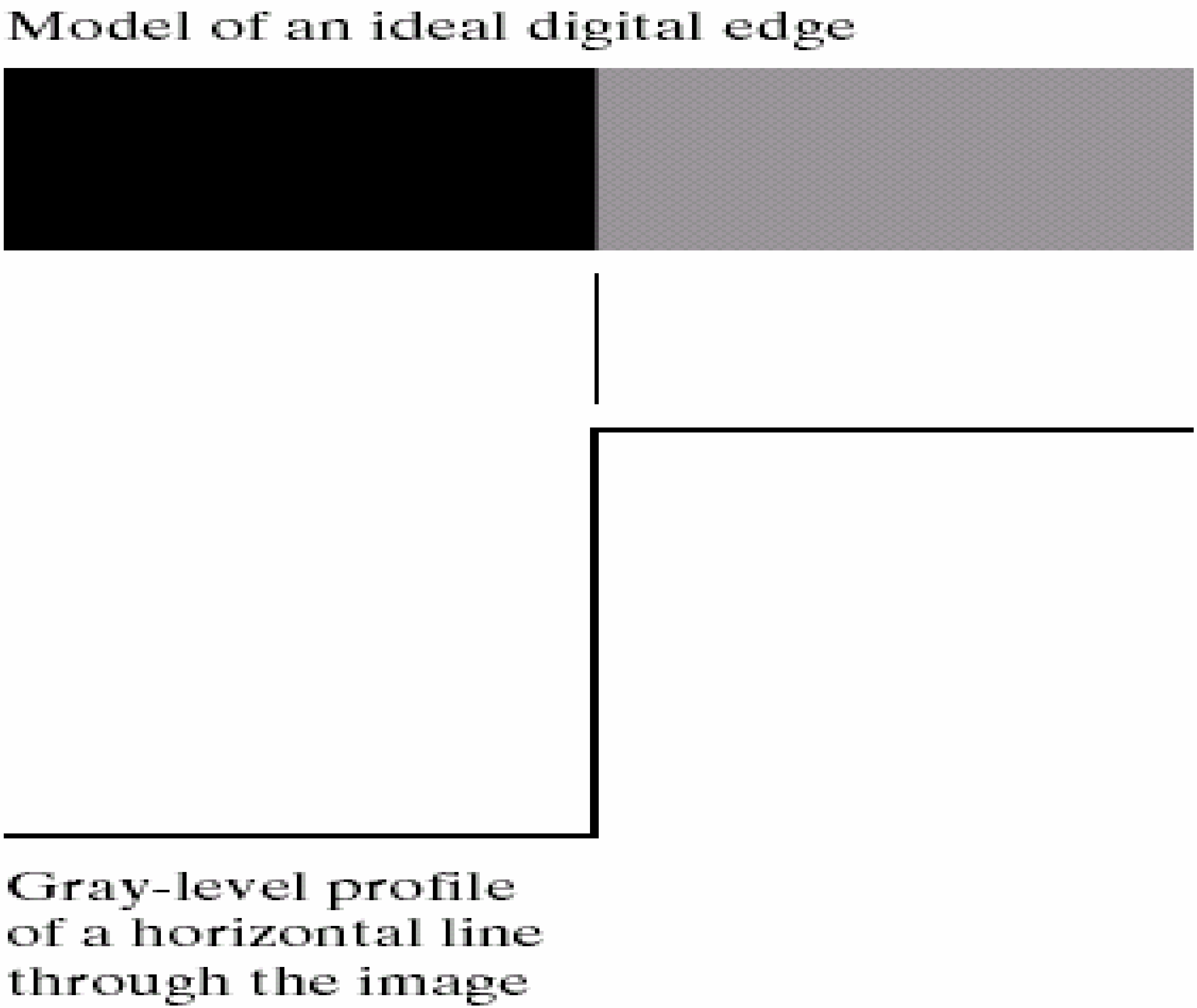
- Points in an image where brightness changes abruptly are called edges or edge points.
- Edges are significant local changes of intensity in an image.
- Edges are the boundaries between segments.
- Edges are a very important portion of the perceptual information content in an image.
- Edges can be broadly classified in to
  - Step edge
  - Line edge
  - Ramp edge
  - Roof edge.

Classification of Edges

- Step Edge
- The Step edge defines a perfect transition from one segment to another.
- In the case of a step edge, the image intensity abruptly changes from one value to one side of the discontinuity to a different value on the opposite side.
- If segments are piecewise constant and pixels can only belong to one segment, then a step edge model is implicitly used.

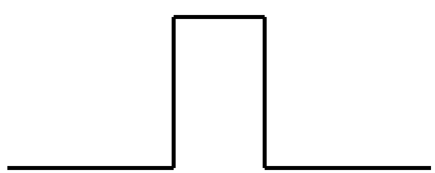


Step edge


Model of an ideal digital edge
Gray-level profile
of a horizontal line
through the image

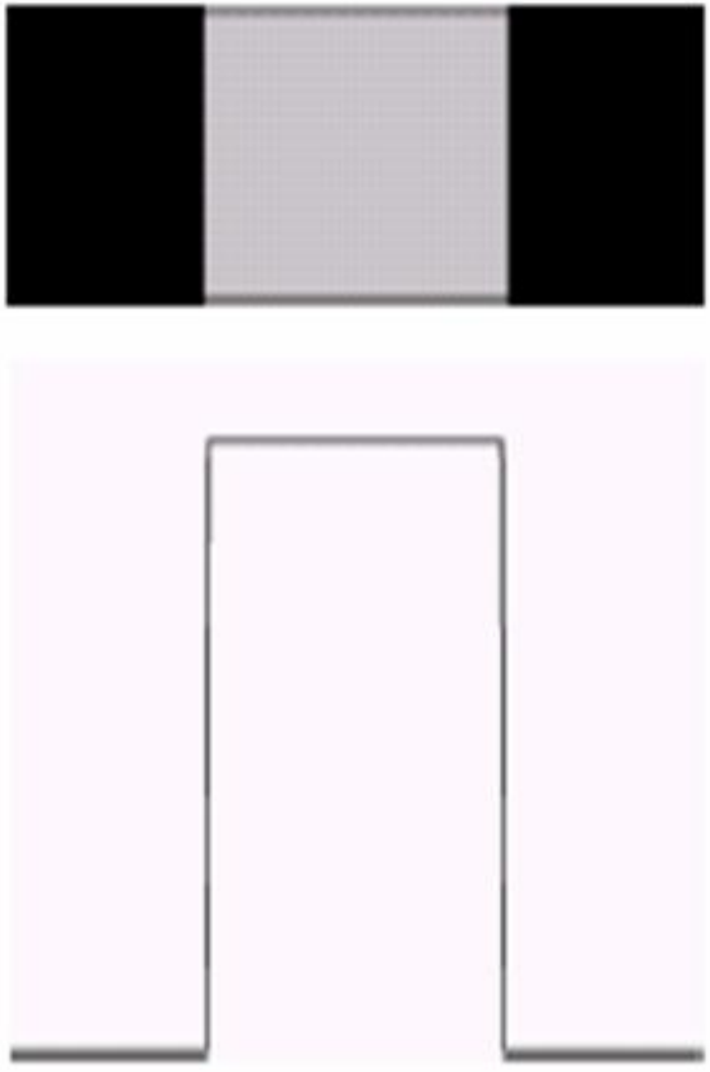
Classification of Edges

- Line Edge
- If a segment of an image is very narrow, it necessarily has two edges in close proximity.
- This arrangement is called a line. The line edge is shown in fig.



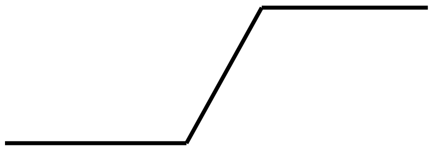
Line Edge

Line Edge



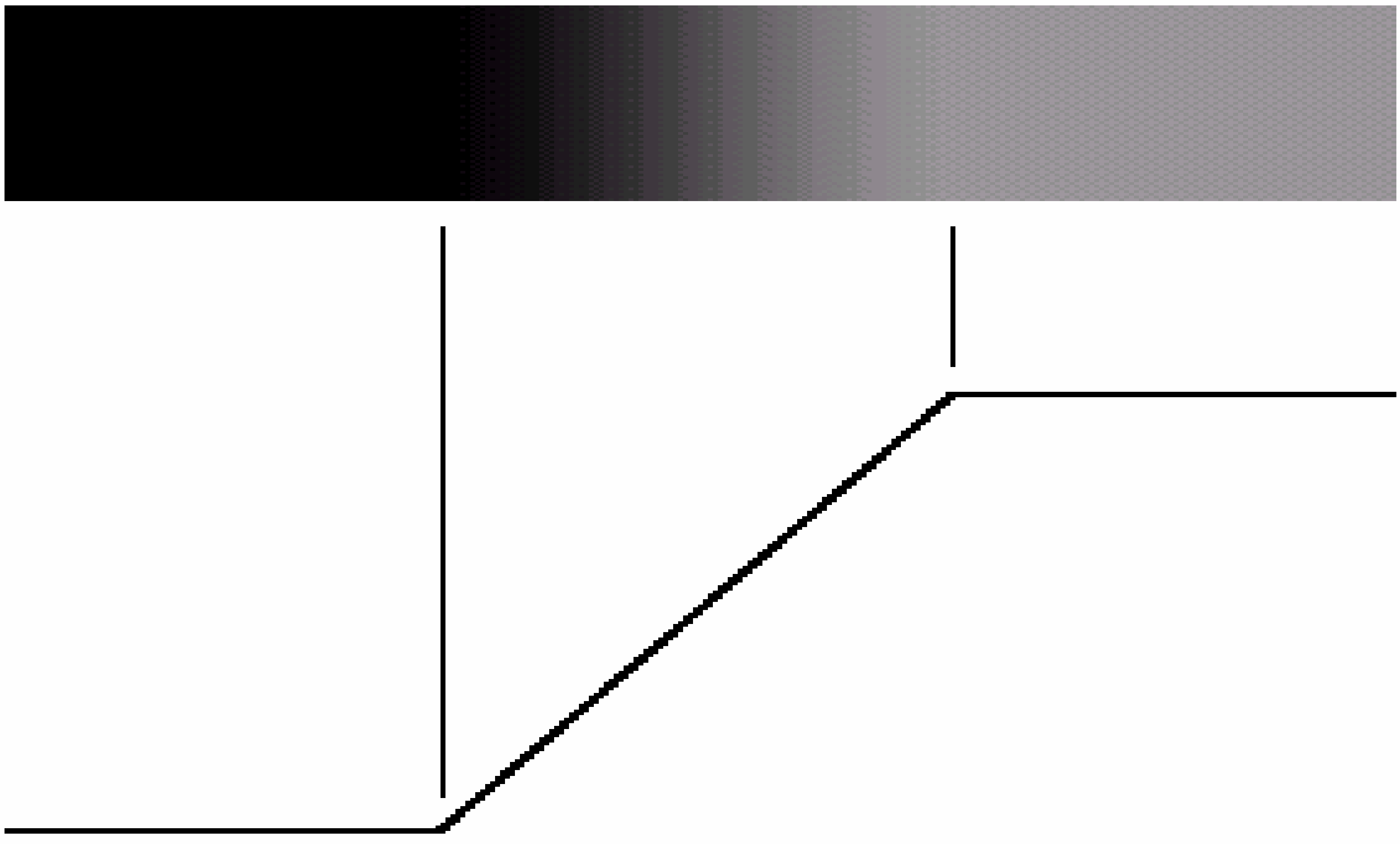
Classification of Edges

- Ramp Edge
- A ramp allows for a smoother transition between segments.
- A ramp edge is useful for modelling the blurred edges created from sampling a scene containing objects not aligned to the pixel grid.
- The ramp edge is shown in following figure.



Ramp Edge

Model of a ramp digital edge

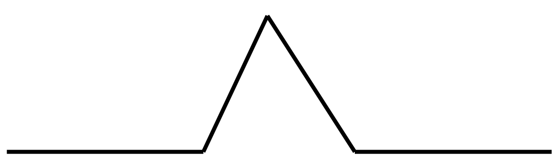


Gray-level profile of a horizontal line through the image

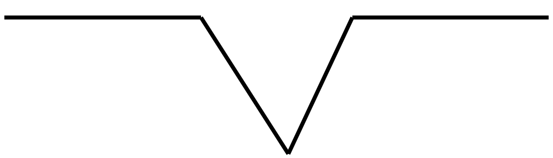
Classification of Edges

❖ Roof Edge

- Two nearby ramp edges result in a line structure called a roof.
- Basically, there are two types of roof edges (i) convex roof edges, and (ii) concave roof edges which are shown in fig.

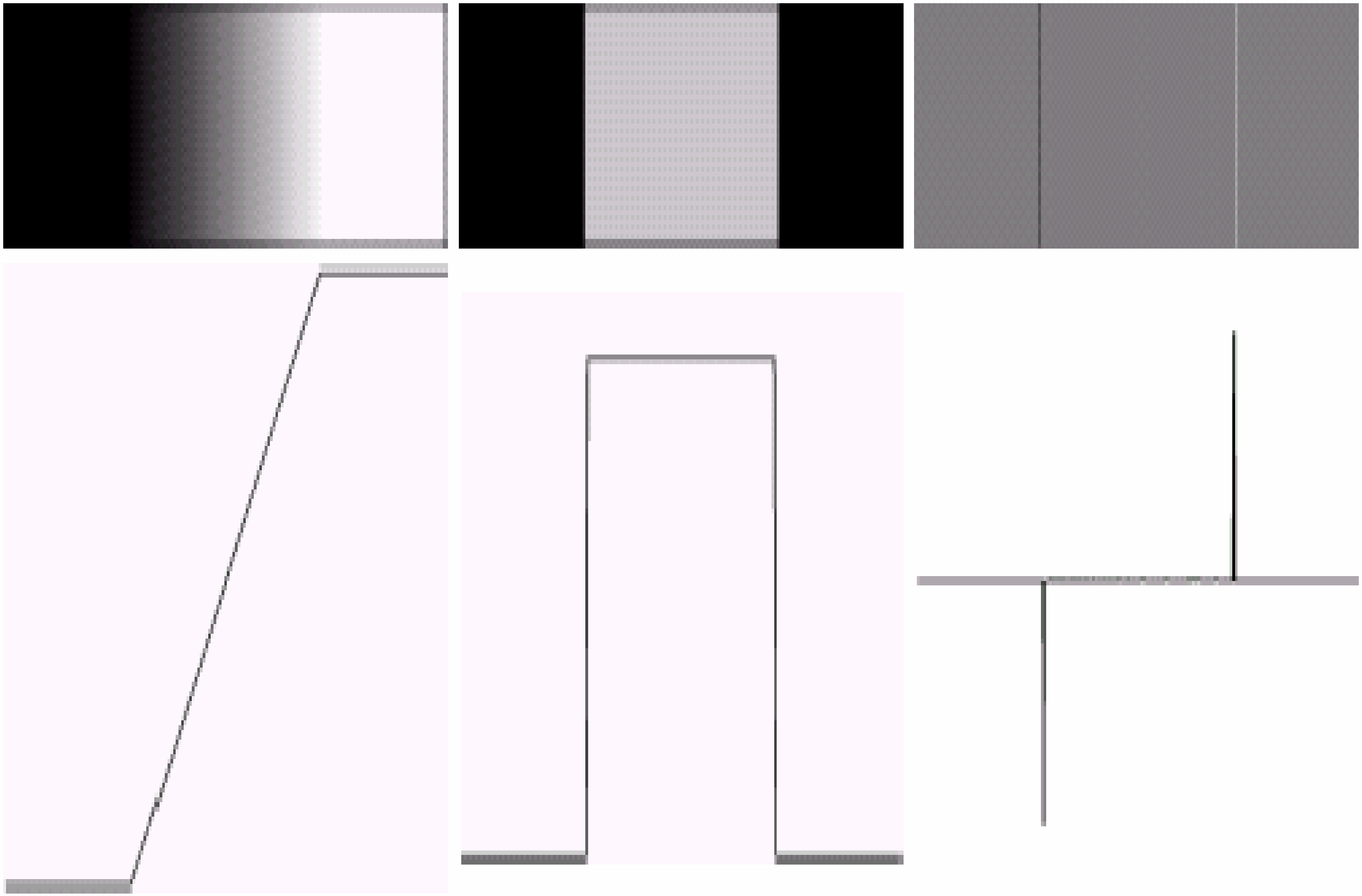


Convex roof edge



Concave roof edge

Different Edges-Example

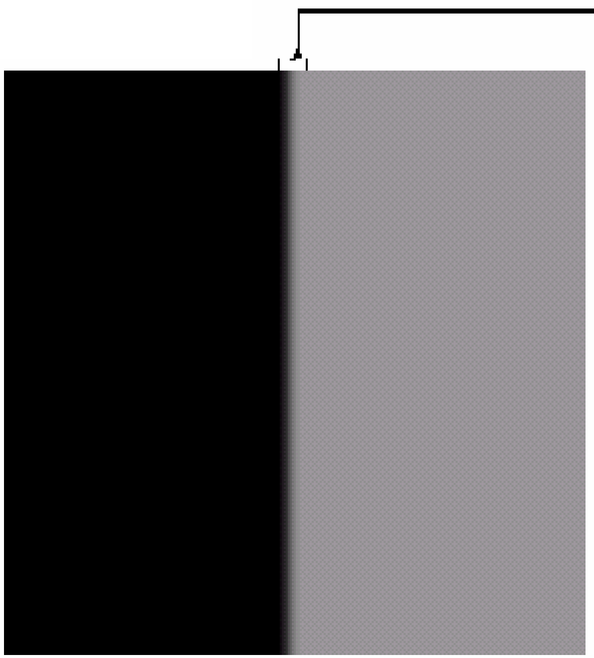


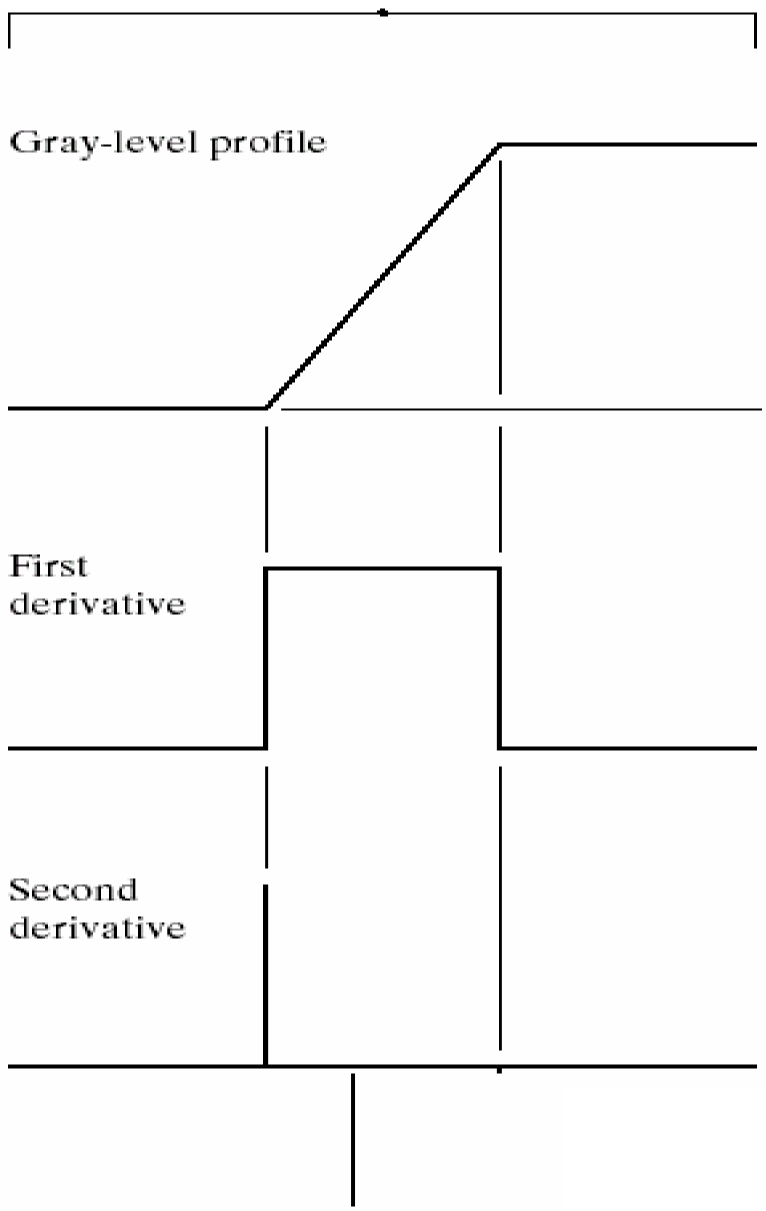
Edge Detection

- Edge detection is the approach of detecting meaningful discontinuities in gray level.
- It is because isolated points and thin lines are not frequent occurrences in most practical applications.
- An edge is the boundary between two regions with relatively distinct gray level properties.
- The regions are assumed to be sufficiently homogenous so that the transitions between the two regions can be determined on the basis of gray level discontinuities alone.

Edge Detection

- The idea behind most effective edge detection techniques is the computation of a ‘local derivative’ operator.
- The first derivative at any point of an image is obtained by using the magnitude of the gradient at that point.
- The second derivative is similarly obtained by using the Laplacian.




Gray-level profile
First derivative
Second derivative

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.
- Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.
- In the previous section, we saw that image blurring could be accomplished in the spatial domain by pixel averaging in a neighbourhood.
- Since averaging is analogous to integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation.
- Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values.

Spatial Sharpening Filters

- Some of the fundamental properties of First order and Second order derivatives in a digital context.

- **First Order Derivative Filter**
  - Must be zero in the areas of flat segment /constant gray levels.
  - Non zero at the onset of gray level step or ramp.
  - Non zeros along ramp.
- **Second Order Derivative Filter**
  - Must be zero in Flat areas.
  - Non zero at onset and end of a gray level step or ramp.
  - Zero along ramps on constant slope.

Spatial Sharpening Filters

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

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

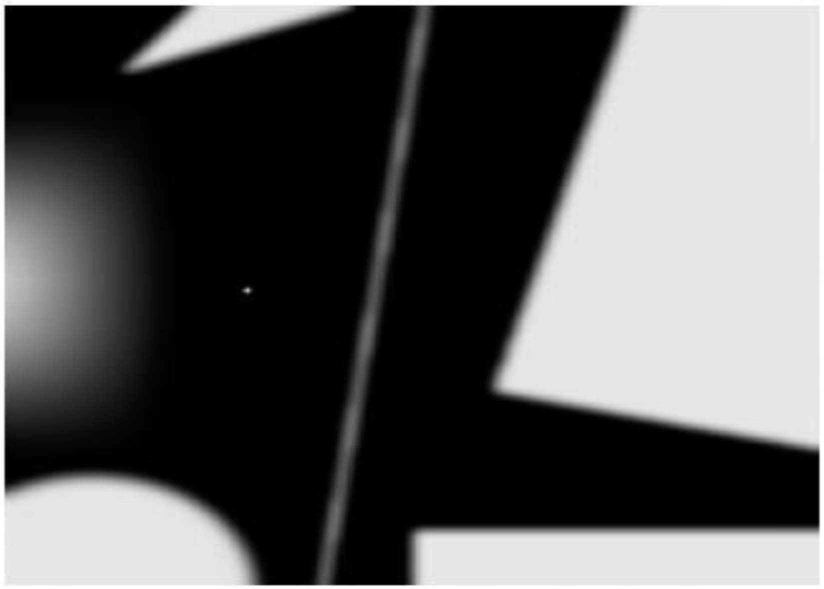
- We used a partial derivative here in order to keep the notation the same as when we consider an image function of two variables, f(x, y).
- Similarly, we define a second-order derivative as the difference.

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

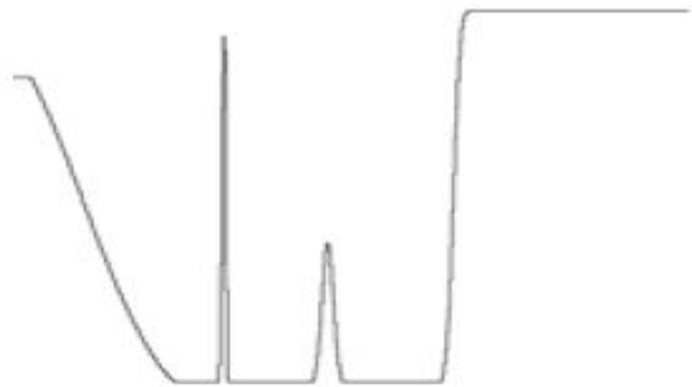
- It is easily verified that these two definitions satisfy the conditions stated previously regarding derivatives of the first and second order.

Spatial Sharpening Filters –Derivative Filter

❖ To highlight the fundamental similarities and differences between first- and second-order derivatives in the context of image processing, consider the example.



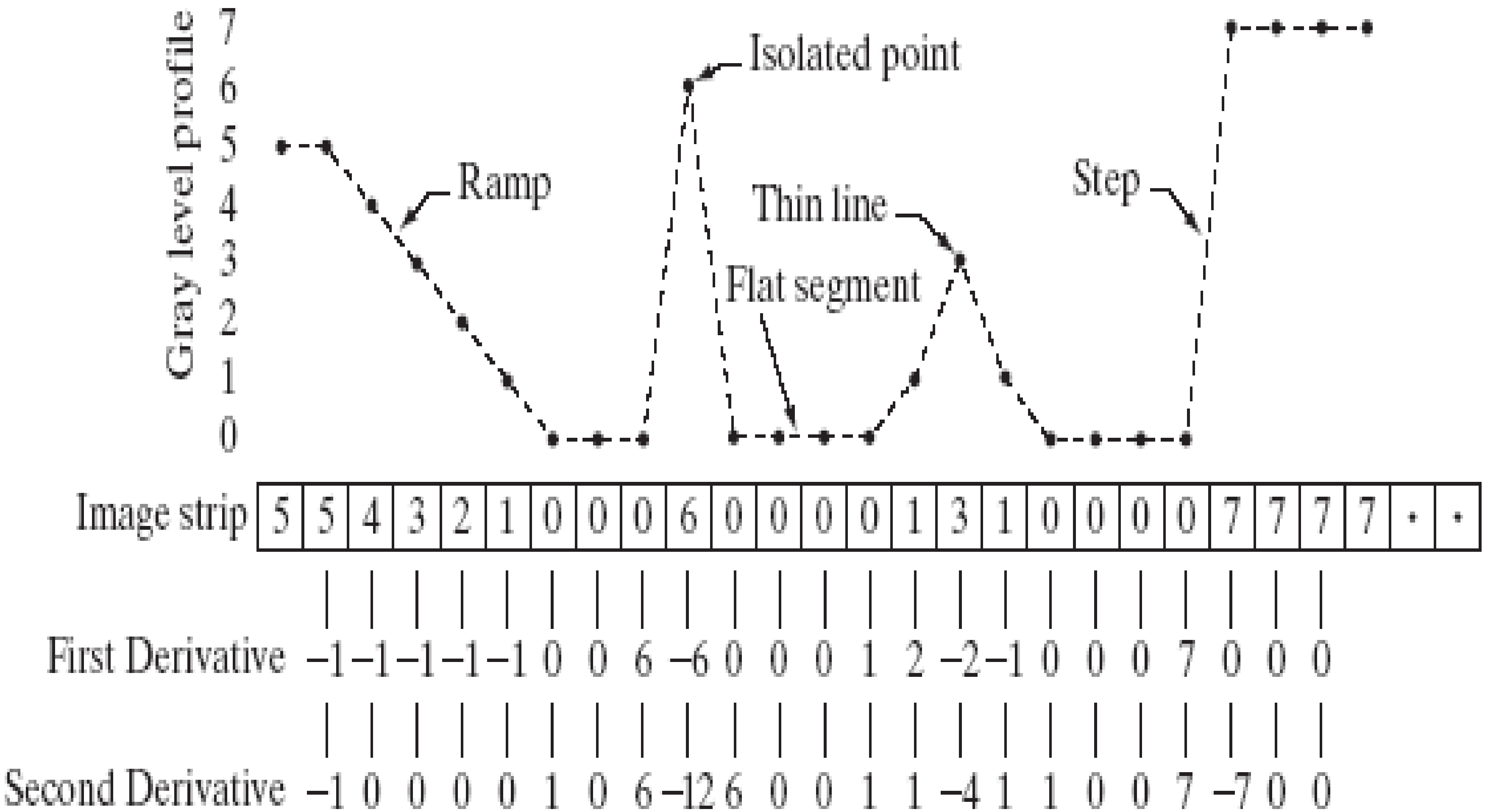
(a)



(b)

(a) A simple image. (b) 1-D horizontal graylevel profile along the center of the image and including the isolated noise point.

First and Second order Derivatives Response



Observations

- First order derivative in non zero along the entire ramp, while the second order derivative is non zero only at the onset and end of the ramp.
- First order derivative generally produce thicker edges in an image.
- Second order derivatives give stronger response than first order derivative to fine details such as thin lines and isolated noise points.
- First order derivative have stronger response to gray level step.
- Second order derivative produce a double response at step edges.
- Second order derivative is better suited for Image Enhancement because of the ability of the former to enhance fine detail..

Laplacian Operator

- The second order derivative of isotropic nature is called as Laplacian operator. Let us consider dot product of ∇

$$\nabla . \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \\ \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \\ \frac{\partial}{\partial y} \end{bmatrix}$$

$\nabla . \nabla = \nabla^2$ is called as the Laplacian Operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad \text{(I)}$$

Laplacian Operator

The Laplacian operator can be expressed in term of difference equation as given below

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

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

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

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

By using equation (I)

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

$$\nabla^2 f= f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)-4f(x,y)$$

Laplacian Mask

The 3 x 3 Laplacian Operator is given by

0	1	0
1	−4	1
0	1	0

(a)

1	1	1
1	−8	1
1	1	1

(b)

(a)Filter mask used to implement the digital Laplacian,
(b) Mask used to implement an extension of this equation that includes the diagonal neighbour's.

Laplacian Mask

The 3 x 3 Laplacian Operator is given by

0	−1	0
−1	4	−1
0	−1	0

(a)

−1	−1	−1
−1	8	−1
−1	−1	−1

(b)

Laplacian Operator

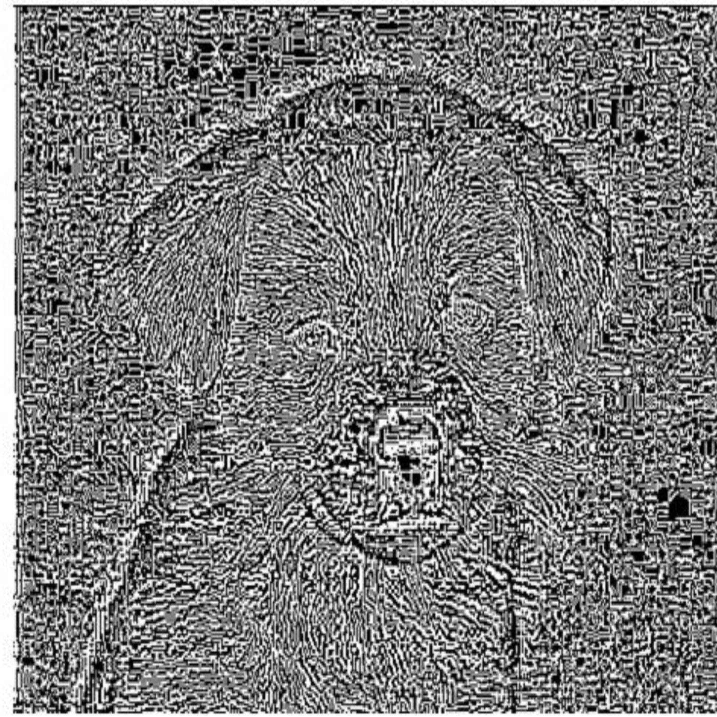
- The Laplacian operator subtracts the brightness values of each of the neighbouring pixels from the central pixel.
- When the discontinuity is present within the neighbourhood in the form of a point, line or edge, the result of the Laplacian is non zero value.
- It may be either positive or negative depending where the central point lies with respect to the edge.
- The Laplacian operator is rotationally invariant. The Laplacian operator does not depend on direction as long as they are orthogonal.

Draw backs of the Laplacian Operator

- Useful directional information is not available by the use of a Laplacian operator.
- The Laplacian, being an approximation to the second derivative, doubly enhances any noise in the image.

Image Sharpening

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



Original Image

Edge Enhanced Image

Image Enhancement by using Laplacian Operator

Figure a

Figure b

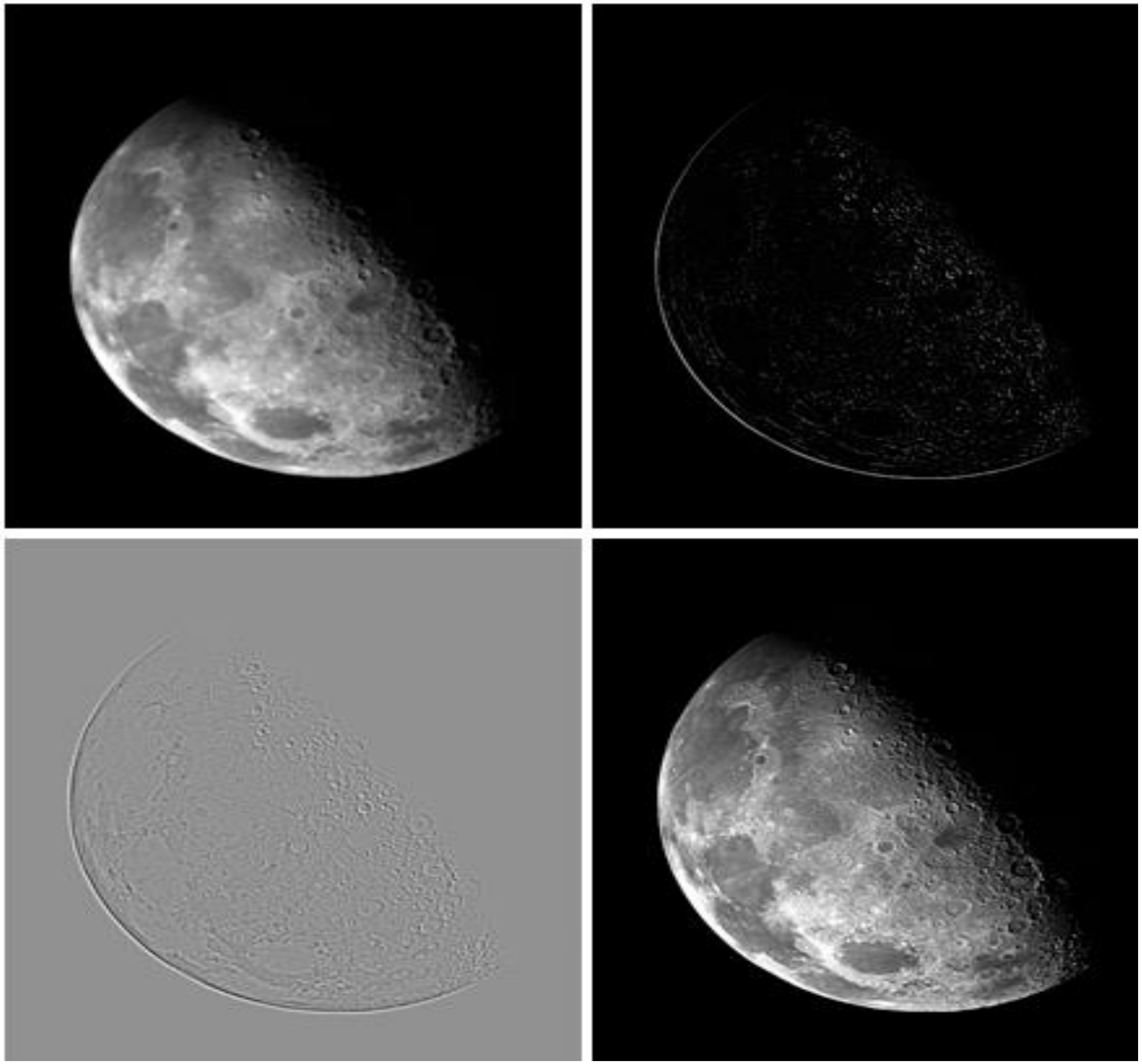


Figure c

Figure d

(a) Image of the North Pole of the Moon. (b) Laplacian Filtered Image
(c) Laplacian Image Scaled for display Purposes. (d) Enhanced Image

Use of the Laplacian for image enhancement

- To simplify the computation we can create the mask which do both the operations i.e. Laplacian filter+ addition operation.

Composite Mask

Background features can be “recovered” while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images.

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

$$\begin{aligned} g(x,y) &= f(x,y) - [f(x+1,y) + f(x-1,y) \\ &\quad + f(x,y+1) + f(x,y-1)] + 4f(x,y) \\ &= 5f(x,y) - [f(x+1,y) + f(x-1,y) \\ &\quad + f(x,y+1) + f(x,y-1)]. \end{aligned}$$

Composite Mask

0	−1	0
−1	5	−1
0	−1	0

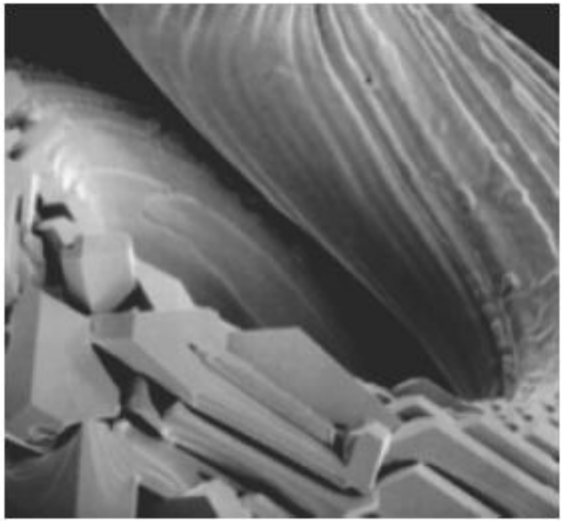
(a)

−1	−1	−1
−1	9	−1
−1	−1	−1

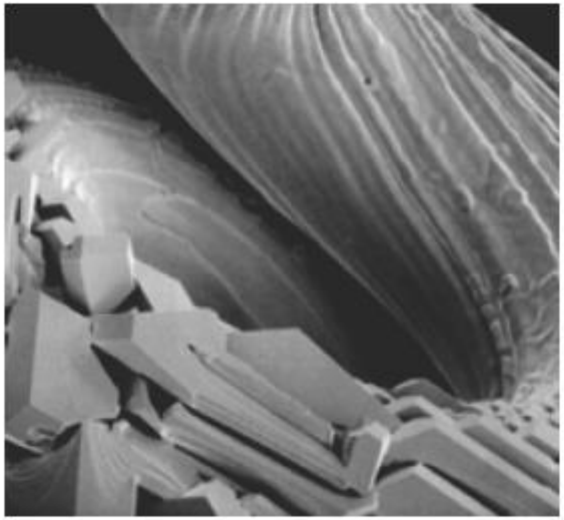
(b)

(a) Composite Laplacian mask (b) A second composite mask

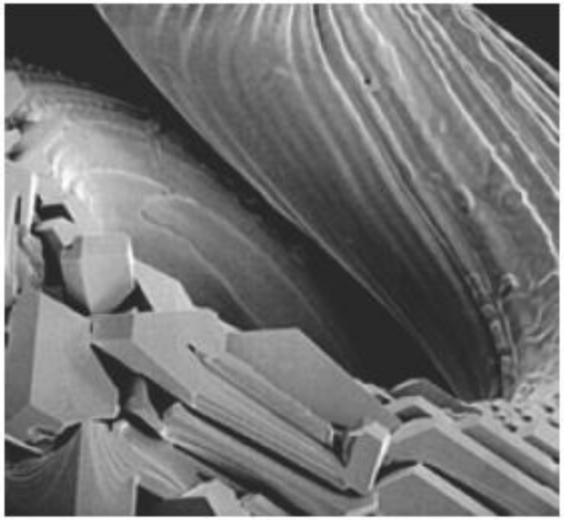
Application-Composite Mask



(a)



(b)



(c)

(a) Scanning Electron microscope image a tungsten. (b) & (c)Result of filtering with masks composite Laplacian mask and Second composite mask

Unsharp masking and high-boost filtering

- A process used in industry to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called unsharp masking is given by

$$f_s(x,y) = f(x,y) - \overline{f(x,y)} \quad \text{--(1)}$$

Where $f_s(x,y)$ denotes the sharpened image obtained by unsharp masking, and $\overline{f(x,y)}$ is blurred version of $f(x,y)$.

A slight further generalization of unsharp masking is called high-boost filtering . A high boost filtered image f_{hb} is defined at any point (x,y) as given below.

$$f_{hb(x,y)} = Af(x,y) - \overline{f(x,y)} \quad \text{--(2)}$$

Where $A \geq 1$

Unsharp masking and high-boost filtering

- Equation can be written in following way also

$$f_{hb(x,y)} = (A-1)f(x,y)+f(x,y)- \overline{f(x,y)}$$

From equation (1)

$$f_{hb(x,y)} = (A-1)f(x,y)+f_s(x,y)$$

- If we are using the Laplacian operator as $f_s(x,y)$ then above equation becomes

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

Where, $f_S(X,Y) = f(x, y) - \nabla^2 f(x, y)$

Unsharp masking and high-boost filtering

- $f_{hb}(x,y)= (A-1)f(x,y)+f(x,y)-\nabla^2 f(x,y)$

$f_{hb}(x,y)=Af(x,y)-f(x,y)+f(x,y)-[f(x+1)+f(x-1,y)+f(x,y+1)+f(x,y-1)]+4f(x,y)$

$fhb(x,y)=(A+4)f(x,y) - [f(x+1)+f(x-1,y)+f(x,y+1)+f(x,y-1)]$

Similarly We can calculate mask by considering 8-neighbourhood.

$f_{hb}(x,y)=Af(x,y)-f(x,y)+f(x,y)-[f(x+1)+f(x-1,y)+f(x,y+1)+f(x,y-1)+f(x-1,y-1)+f(x-1,y+1)+f(x+1,y-1)+f(x+1,y+1)]+8f(x,y)$

$fhb(x,y)=(A+8)f(x,y)-[f(x+1)+f(x-1,y)+f(x,y+1)+f(x,y-1) +f(x-1,y-1)+f(x-1,y+1)+f(x+1,y-1)+f(x+1,y+1)]$

High Boost Filter

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

The high-boost filtering technique can be implemented with either one of these masks.

❑ **Note that**

❖ When A=1, high boost filtering becomes "standard" Laplacian sharpening.

❖ As the value of A increases past 1, then the contribution of the sharpening process becomes less and less important.

❖ Eventually, if A is large enough, the high-boost image will be approximately equal to the original image multiplied by a constant.

❖ One of the principal application of boost filtering is when the input image is darker than desired. By varying the boost coefficient, it generally possible to obtain an overall increase in average gray level of the image.

Use of the First Derivatives for Enhancement-The Gradient

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

- $$\nabla f = \begin{bmatrix} Gx \\ \text{Gy} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Use of the First Derivatives for Enhancement-The Gradient

The Magnitude of this vector is given by

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

The Computational burden of implementing above equation over an entire image is not trivial and it is common practice to approximate the magnitude of the gradient by using absolute values instead of squares and squares roots

$$\nabla f = |G_x| + |G_y| \text{ ------(1)}$$

This equation is simpler to compute and it still preserve relative changes in gray levels.

Use of the First Derivatives for Enhancement-The Gradient

- The simplest approximation to a first order derivative are

$G_x = (Z_8 - Z_5)$ and $G_y = (Z_6 - Z_5)$

Two other equations proposed by Roberts in the early development of digital image processing use cross differences

$G_x = (Z_9 - Z_5)$ and $G_y = (Z_8 - Z_6)$

We compute the gradient as

$$\nabla f = [(Z_9 - Z_5)^2 + (Z_8 - Z_6)^2]^{1/2}$$

If we use absolute values, then by using equation (1)

$$\nabla f \cong |Z_9 - Z_5| + |Z_8 - Z_6|$$

This equation can be implemented with the two masks. These masks are referred to as the Robert cross gradient operators.

Use of the First Derivatives for Enhancement-The Gradient

Z_1	Z_2	Z_3
Z4	Z_5	Z_6
Z_7	Z_8	Z_9

Figure a

Figure b

-1	0
0	1

Figure c

0	-1
1	0

Figure a denotes image points in a 3x3 region. Figure b and figure c, these masks are referred to as the Robert cross gradient operators

First Order Derivative-Sobel Operators

- Masks of even size are awkward to implement. The smallest mask which we are interested is of size 3x3.

$$\nabla f \approx |(Z_7+2Z_8+Z_9) - (Z_1+2Z_2+Z_3)| + |(Z_3+2Z_6+Z_9) - (Z_1+2Z_4+Z_7)|$$

- The difference between the third and the first row rows of the 3x3 image region approximates the derivative in the x-direction.
- The difference between third and first columns approximates the derivatives in the y-direction.
- These mask are called as Sobel operators.

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

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

First Order Derivative-Sobel Operators

- $$\frac{\partial f}{\partial x}$$

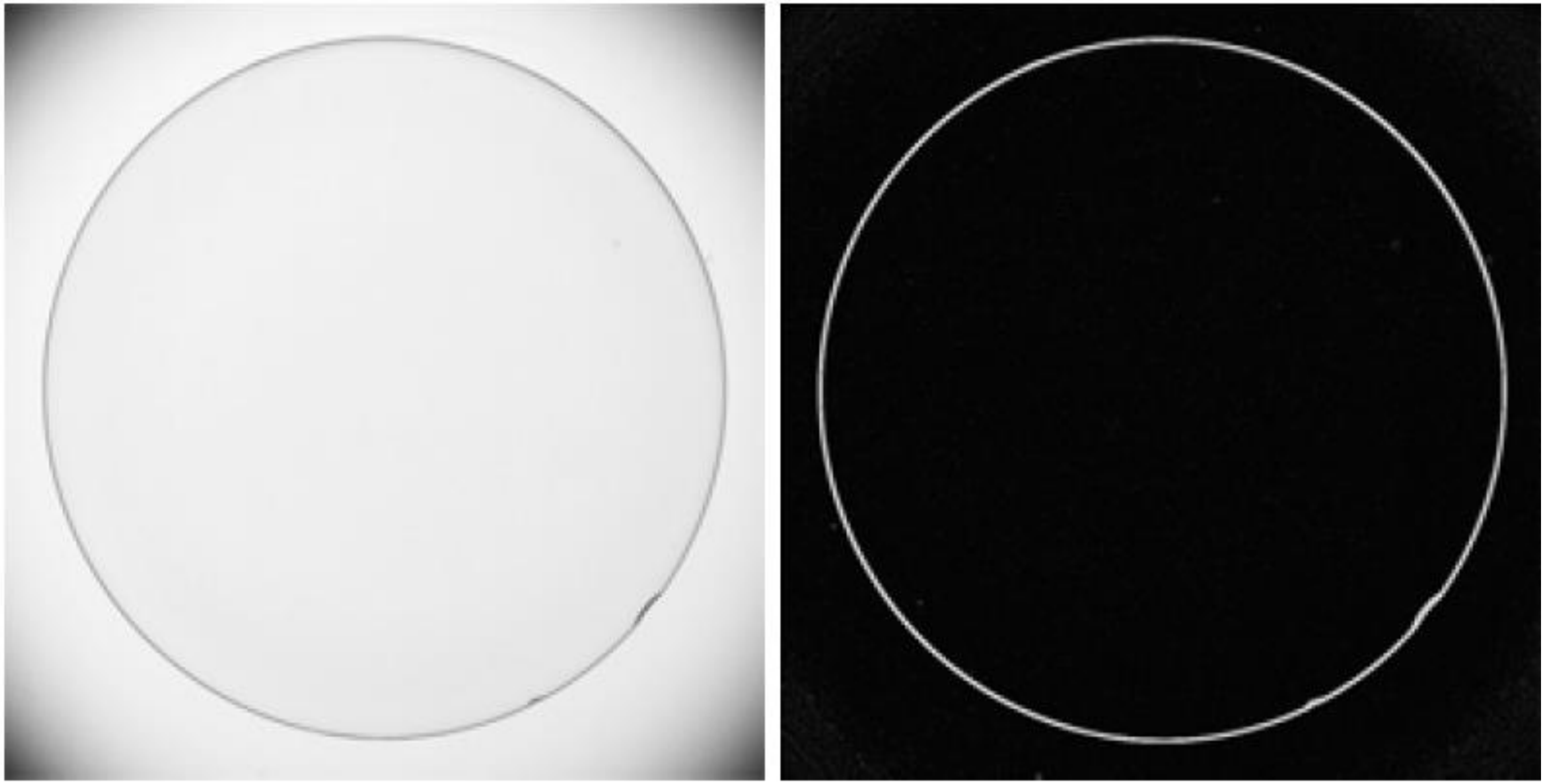
-1	-2	-1
0	0	0
1	2	1

$$\frac{\partial f}{\partial y}$$

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

- The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point.
- Note that the coefficient in all above mask shown, sum to 0, indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator.

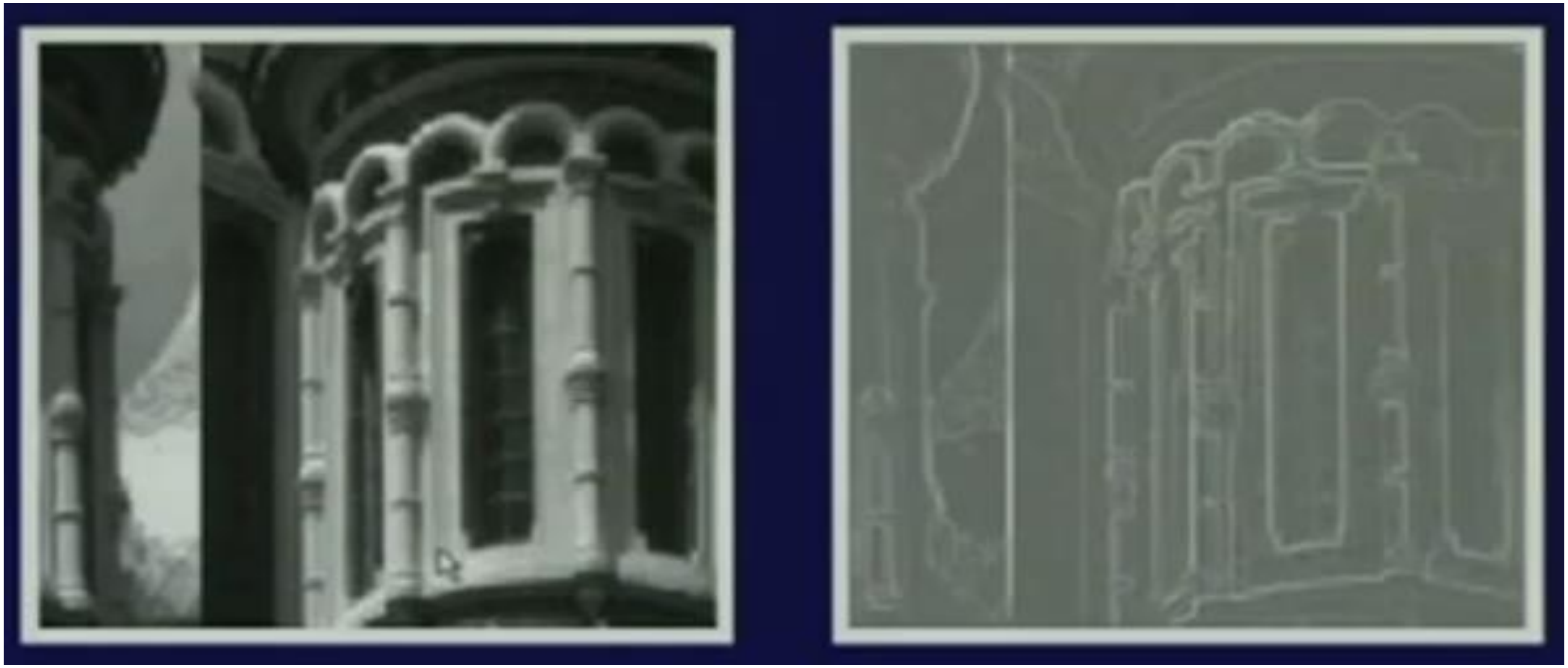
Application –Sobel Operators



(a) (b)

(a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.

First Order Derivative Example



Thanks

Stereo Vision

By,
Prof(Dr) Premanand P Ghadekar

Outline

- Introduction to Stereo Vision
- Triangulation
- Stereo Imaging
- Stereo Vision Terminology
- Two Main Problems of Stereo Vision

Stereo Vision

- We use our left and right eyes to capture two images of the same scene or object.
- Both images give different viewpoints of the same scene.
- However, we do not perceive the scene as two images. Rather we perceive a single fused image of a scene.
- The human visual system uses a **slight difference in the left and right images to estimate the distance between the eye and the points on the object.**
- It then obtains **the depth information from the two images captured by both the eyes.**
- The human eye also calculates **the angle of convergence, which is the amount of eye rotation about the vertical axis, which fuses the left and right images to recreate a 3D world for us.**

Stereo Vision

- Stereoscopic imaging or stereo imaging **aims to emulate the human visual system to create a 3D vision.**
- It obtains **depth information necessary to create 3D objects from multiple 2D images**, similar to the use of 1D projection data to create a 3D CT image.
- Stereo Imaging **can use multiple cameras** to capture a scene.
- A simple model having two cameras is know as **Binocular Imaging**. The first requirement is that the cameras should be calibrated.
- **Camera Calibration** means obtaining the values of camera parameters to establish a connection between the points of the real scene and the image. If the cameras are moving, then multiple calibrations must be done.

Stereo Vision

- One-time measurement is sufficient for a static camera.
- In **binocular imaging, two cameras, Similar to the human visual system, capture two separate images.**
- The rays **from left and right eyes form a triangle.** This gives a point in the image plane. This method is called **Triangulation Method.**

What the Stereo Vision Aims

- ❖ **Retrieving 3D information**, and **Structure of an object** with two, or one moving camera.
- ❖ A line and a plane, not including it, intersect in just one point.
- ❖ Lines of sight are easy to compute, and so its easy to tell where any image point projects on to any known plane.
- ❖ If two images from different viewpoints can be placed in correspondence, the intersection of the lines of sight from two matching image points determines a point in 3D space.

Stereo Vision and Triangulation

- One of the first ideas that occurs to one who wants to do three-dimensional sensing is the biologically motivated one of stereo vision.
- Two cameras, or one from two positions, can give relative depth, or absolute three-dimensional location.
- There has been considerable effort in this direction [Moravec 1977, Quam and Hannah 1974, Binford 1971, Turner 1974, Shapira 1974]

The Technique

1. Take two images separated by a baseline

2. Identify points between the two images

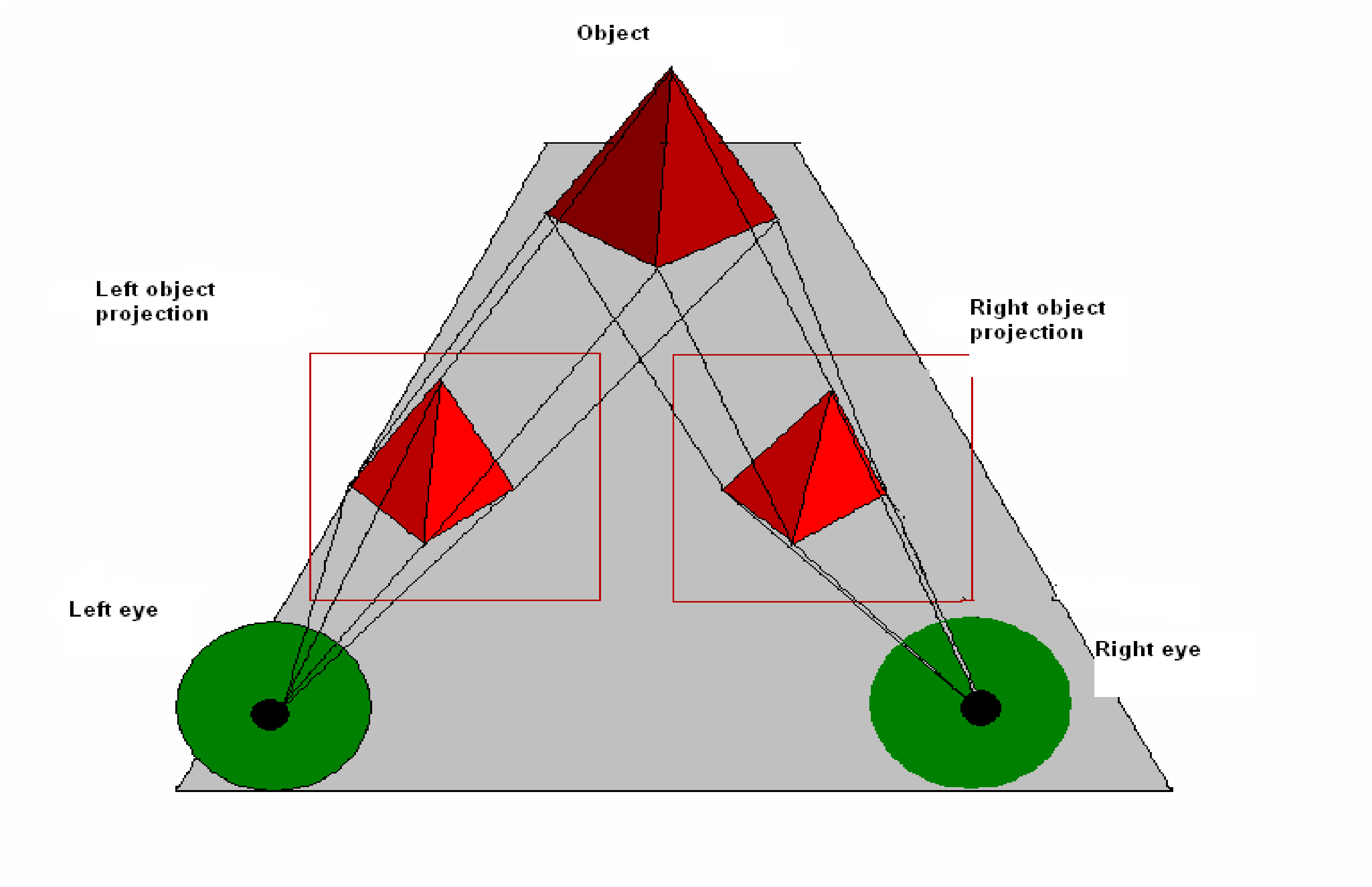
3.Use the inverse perspective transform, or simple triangulation to derive the two lines on witch the world points lie.

4. Intersect the lines

The resulting point is in three-dimensional world coordinates.

- **The hardest part of this is method is step 2, that of identifying corresponding points in the two images.**

Stereo Imaging

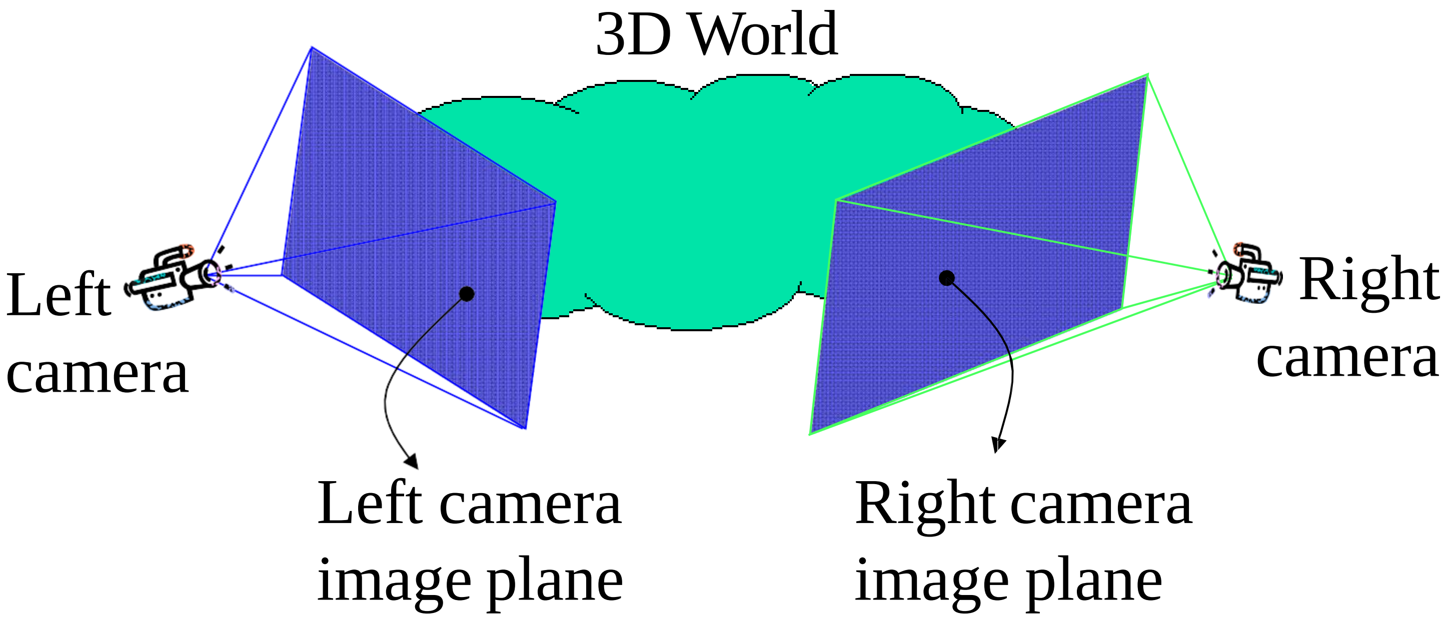


Stereo Vision Terminology

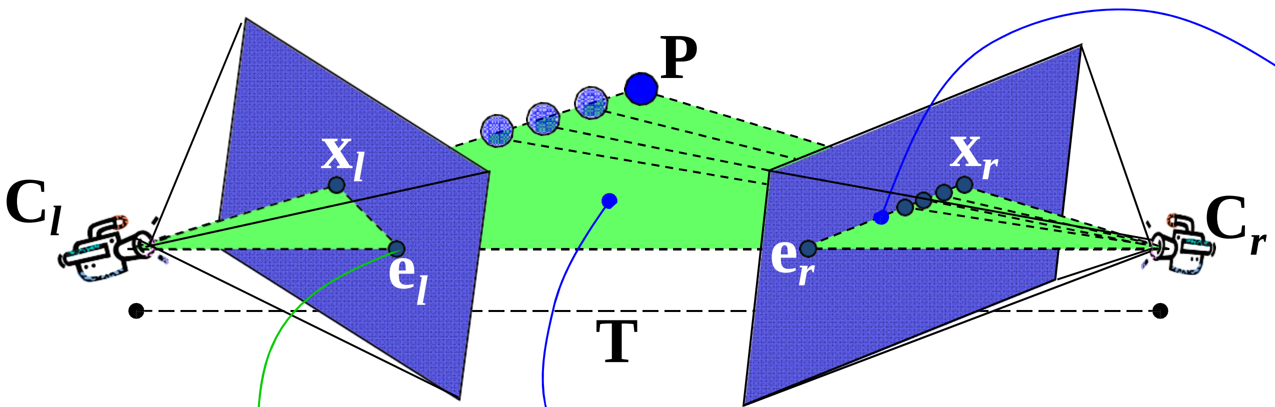
- Fixation point: The intersection of Optical axis
- Baseline: The Distance between the centers of the Projection
- Epipolar: A plane that is defined by the camera center and the object point.
- Epipolar plane: The plane passing through the centers of projection and the point in the scene
- Epipolar line: The intersection of the Epipolar Plane with the image plane
- Conjugate pair: Any point in the scene that is visible in both cameras will be projected to a pair of image points in the two images

Epipolar Geometry

- Defined for two static cameras



Epipolar Geometry



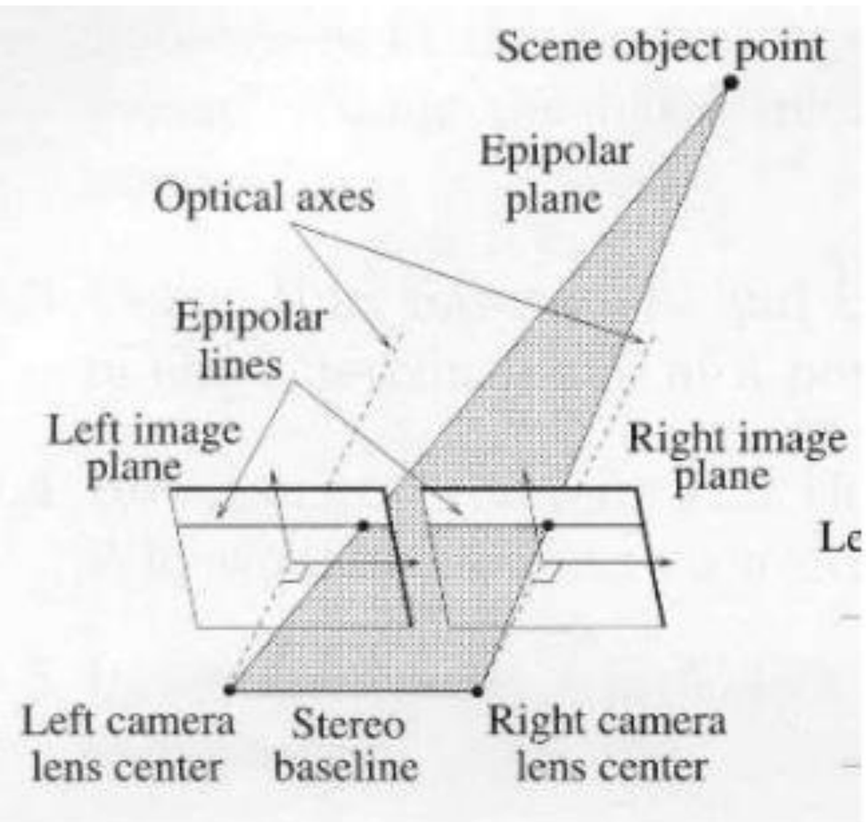
Epipolar line: set of world points that project to the same point in left image, when projected to right image forms a line.

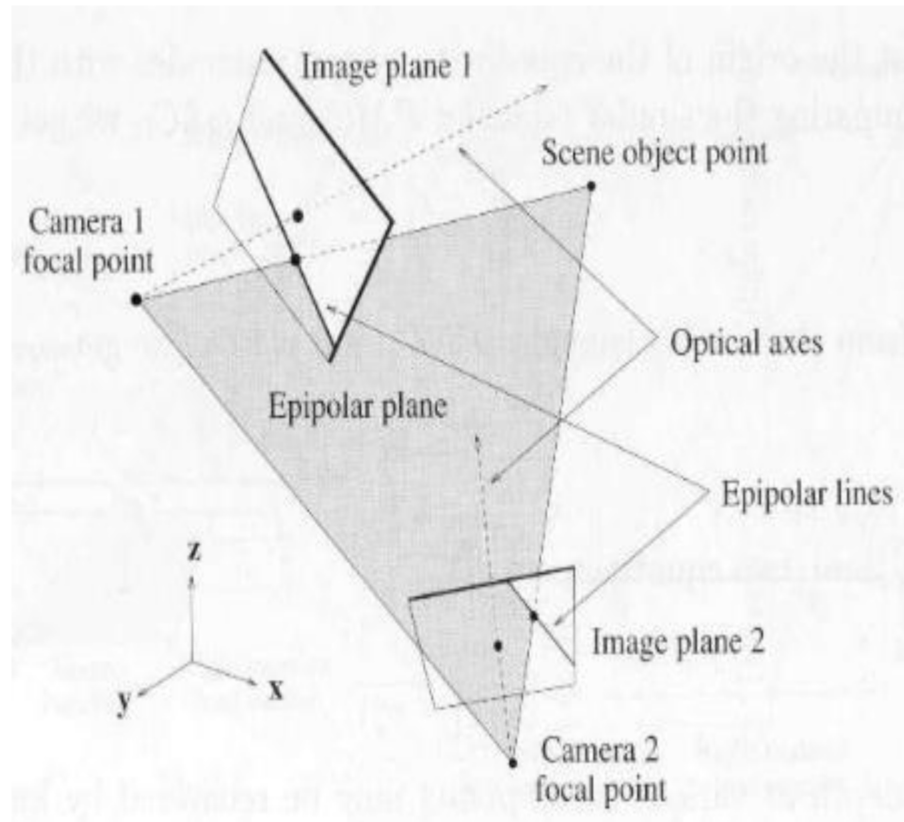
Epipolar plane: plane defined by the camera centers and world point.

Epipole: intersection of image plane with line connecting camera centers. Image of a left camera center in the right, and vice versa.

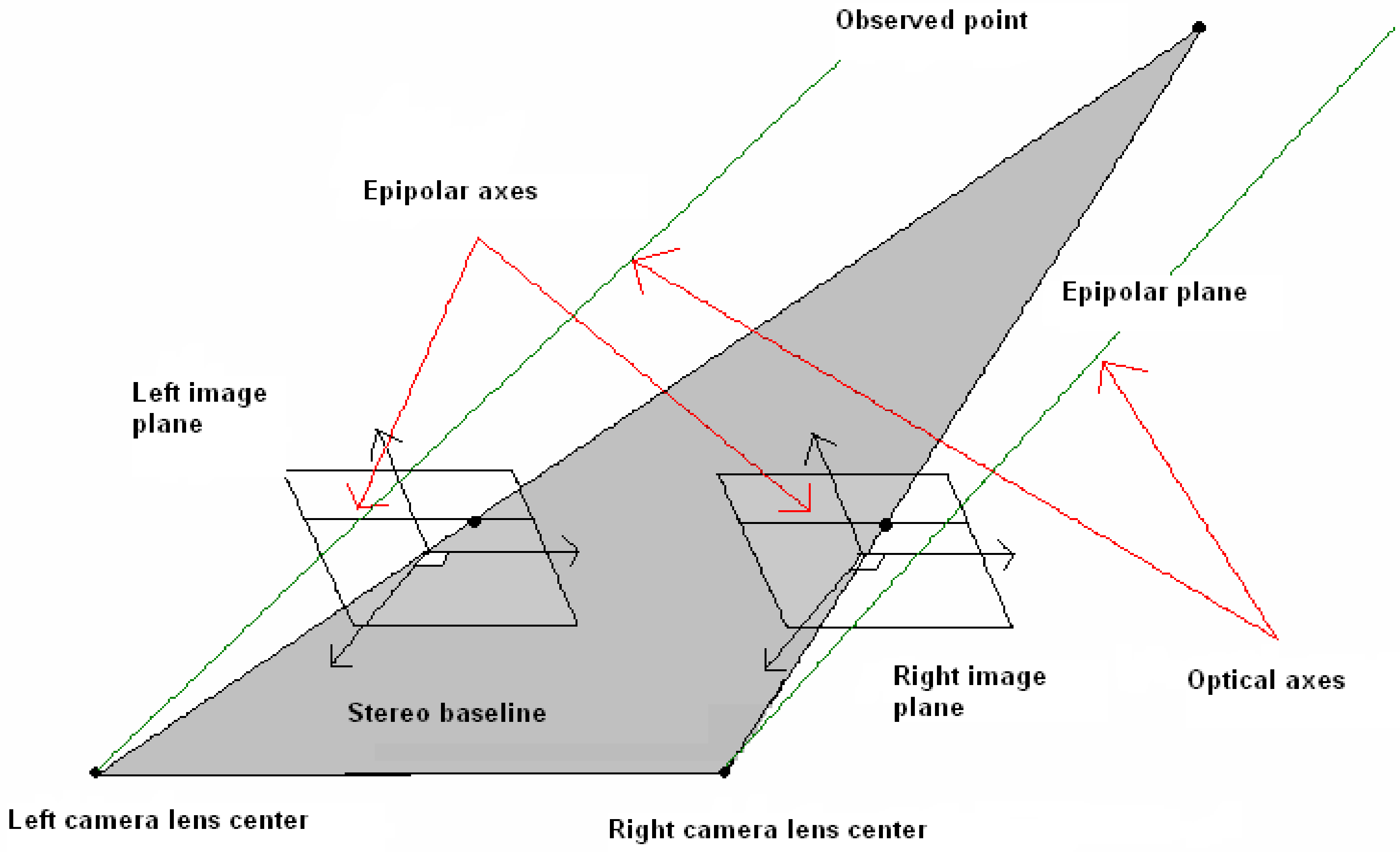
Stereo Vision Terminology

Disparity: The distance between corresponding points when the two images are superimposed
Disparity map: The disparities of all points from the disparity map (can be displayed as an image)





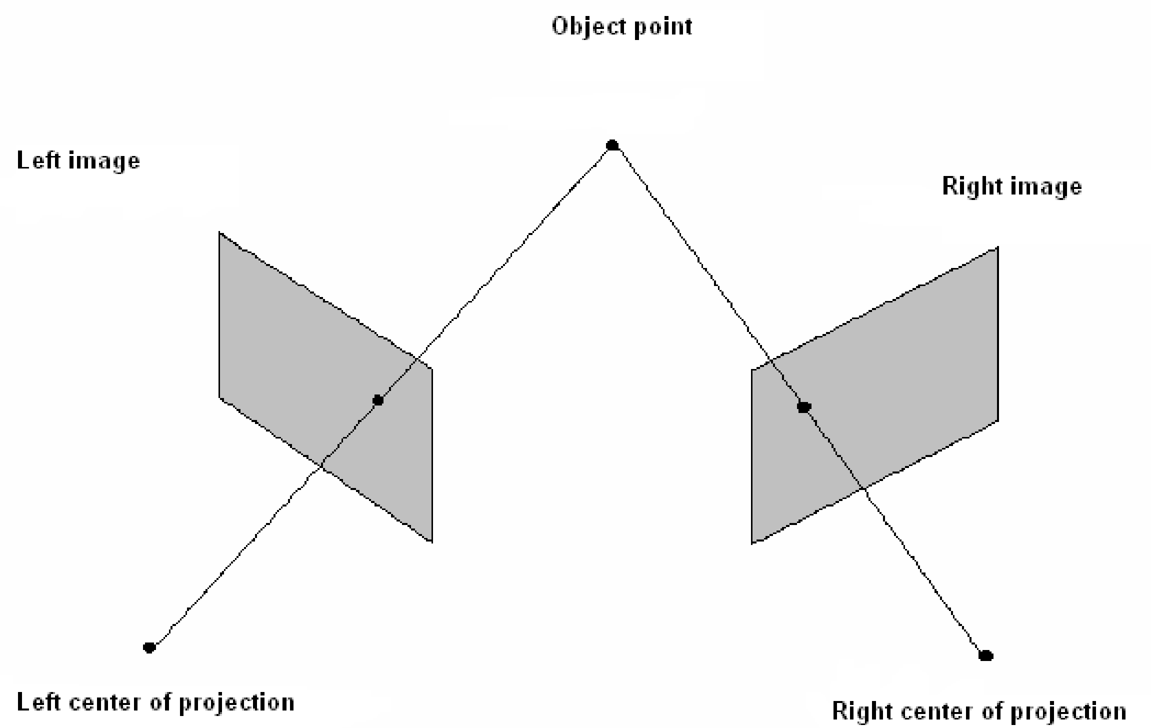
Stereo Imaging



Triangulation-The principle underlying Stereo Vision

- The **3D location** of any visible object point in space is restricted to **the straight line that passes trough the center of projection** and projection of the object point
- **Binocular Stereo Vision** determines the **position of a point in space** by finding the intersection of the two lines passing through the center of projection and the projection of the point in each image.

Triangulation-the Principle underlying Stereo Vision

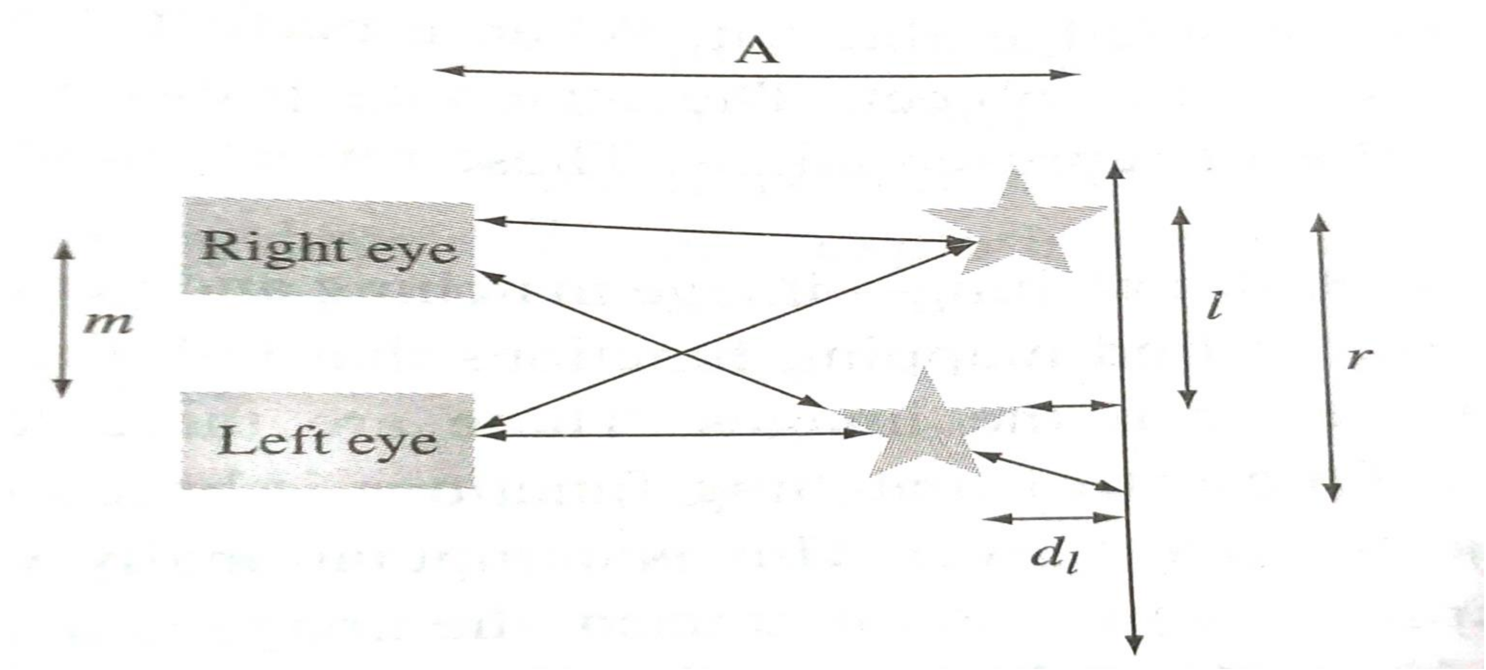


Reconstruction from X

- The next important step of stereo imaging is the **Calculation of Depth information from Disparity Images.**
- This is dependent on many factors such as **Camera Calibration, Feature Detection, and Matching Precision.**
- Stereo vision uses the **Principle of Object Parallax,** which is a measure of how the objects relate to each other spatially when viewed from different viewpoints.
- The data necessary for depth calculation are **Distance, Eye Separation, Left difference, and Right difference.**
- **'A'** is the distance between the camera and the object, **'m'** is the distance between the two eyes, **'D'** is the distance between the right eye and the star, **'l'** is the left separation, **'r'** is the right separation, and **'d_l'**, is the depth difference. The Depth Can be calculated as below

$$\boxed{\textbf{Depth} = \frac{D}{m}\,(r-l)}$$

Reconstruction from X



- Using the **Depth information**, the images are fused into a single image.
- Post-processing of the fused image involves processing of the used image for depth interpolation, geometric distortion, and noise interference.
- **Stereoscopic Imaging** is one method of obtaining Depth information. Similarly, the other Such as illumination factor, shading, and texture are determined.

Two Main Problems of Stereo Vision

I. The correspondence problem

II. The reconstruction problem

The Correspondence Problem

- The **Correspondence Problem** refers to the problem of ascertaining which parts of one image correspond to which parts of another image, where differences are due to movement of the camera, the elapse of time, and/or movement of objects in the photos.
- **Triangulation depends** crucially on the solution of the **Correspondence Problem**.
- **Ambiguous Correspondence** between points in the two images may lead to **several different consistent interpretation of the scene**

The correspondence problem

- If two or more images of the same 3D scene, taken from different points of view.
- The **correspondence problem** refers to the task of **finding a set of points in one image which can be identified as the same points in another image.**
- To do this, **points or features in one image are matched with the corresponding points or features in another image.**
- The problem is made more difficult when the objects in the scene are in **Motion Relative to the Camera(s).**
- **Relative motion** is defined as the calculation of the motion of an object with regard to some other moving or static object

Why is the correspondence problem difficult?

- Some points in each image will have no corresponding point in the other image, because:
- The cameras may have different fields of view
- Due to occlusion [**Occlusion** often occurs when two or more objects come too close and seemingly merge or combine with each other.]
- A stereo system must be able to determine the image parts that should not be matched.

Methods for Establishing Correspondence

- There are two issues to be considered:
- How to select candidate matches ?
- How to determine the goodness of a match?
- A possible class of algorithm
- Correlation based attempt to establish correspondence by matching image intensities

Correlation-based methods

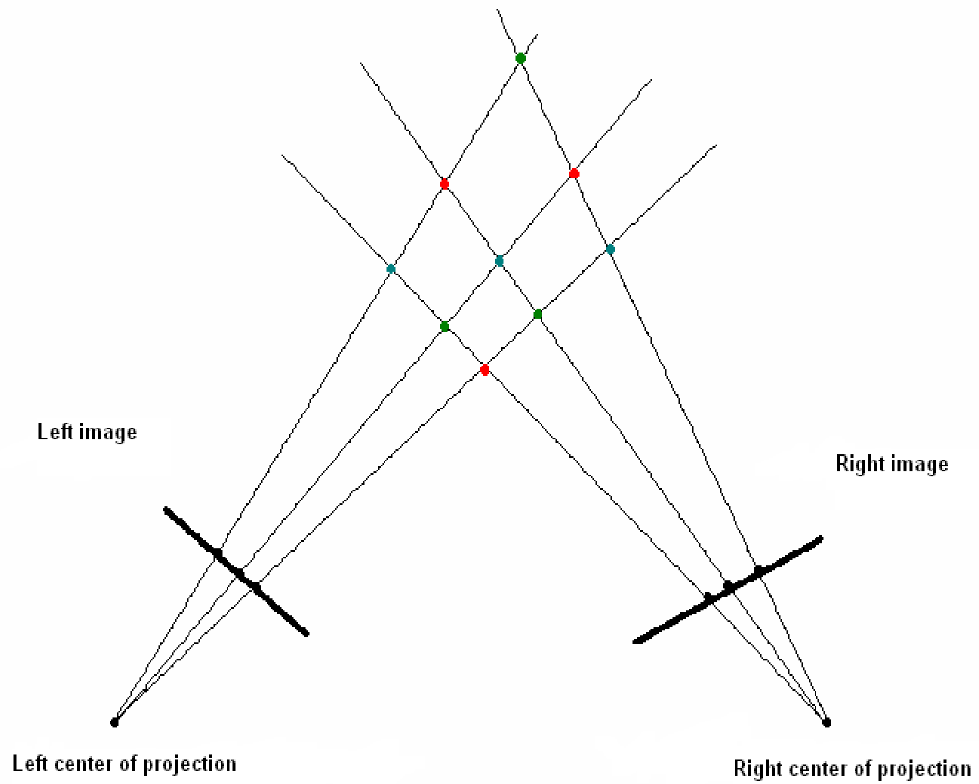
Match image sub-windows between the two images using image correlation

Scene points must have the same intensity in each image (strictly accurate for perfectly matte surfaces only)

The Reconstruction Problem

❖ Given the corresponding points, we can compute the disparity map

❖ The disparity can be converted to a 3D map of the scene



Incorrect matching can give bad results

The Reconstruction Problem

- Once the correct correspondences in the two images are established, the three-dimensional information has to be calculated.
- The disparity changes between the single matches in the images give a relative description of the depth relationships in the environment.
- Which need to be transformed into Cartesian coordinates to allow map generation and navigation.
- This is done using the external (position of the cameras to each other) and internal (focal lengths, radial distortions of the lenses and pixel sizes of the cameras) parameters as a reference.

Stereo Imaging Applications

Stereo Vision Technology is used in a variety of applications, including-

- People Tracking, Mobile Robotics Navigation, and Mining.
- It is also used in Industrial Automation.
- 3D machine vision applications to perform tasks such as bin picking, volume measurement.
- Automotive part measurement and 3D object location and identification.
- Stereo vision is also becoming the technology of choice for range sensing in mobile robotics navigation.
- Robot navigation means the robot's ability to determine its own position in its frame of reference and then to plan a path towards some goal location.


Diolch
Kiitos
Sheun
umesc
Kasih
Todah
Mamnoon
Shnorhakalutiun
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ači
Gamsahapnida
Teşekkür
Dekuju/Dekujeme
Hvala
Ngiyabonga
Xie
Dank
Takk
Cam
Gra
or
Dhanyavaad
Dakujem
Waad
Dhanyavaadalu
Kop
Salamat
Merci
al
Dankie
Go
Grazie
Faleminderit
Daw
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Fyrir
Terima
Enkosi
Danke
dank
Euxaristo
Kun
Hain
Dhan
daa

Wavelet Transform

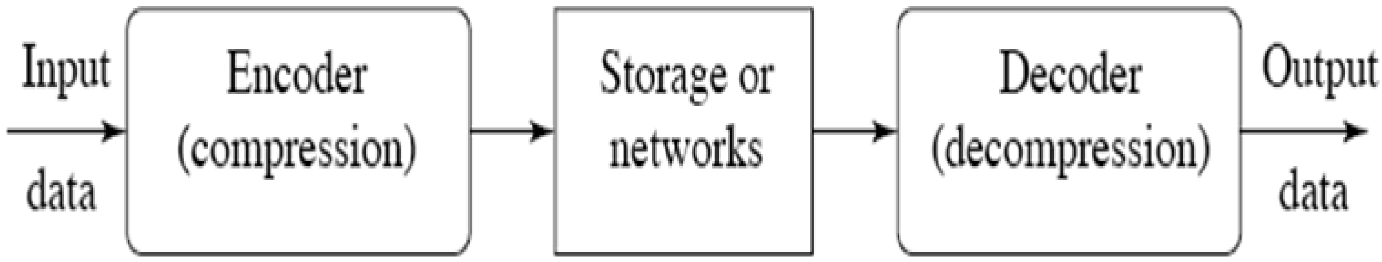
By,
Prof (Dr.) P P Ghadekar

Outline

- On the completion student will be able to understand
- MPEG1 Compression Technique
- Basic concepts of Wavelet transform
- 1-D and 2-D Wavelet transform
- Properties of Wavelet transform
- Sub band coding
- Quality Assessment Parameters

Compression

❖ **Compression:-** The process of coding that will effectively reduce the total number of bits needed to represent certain information.

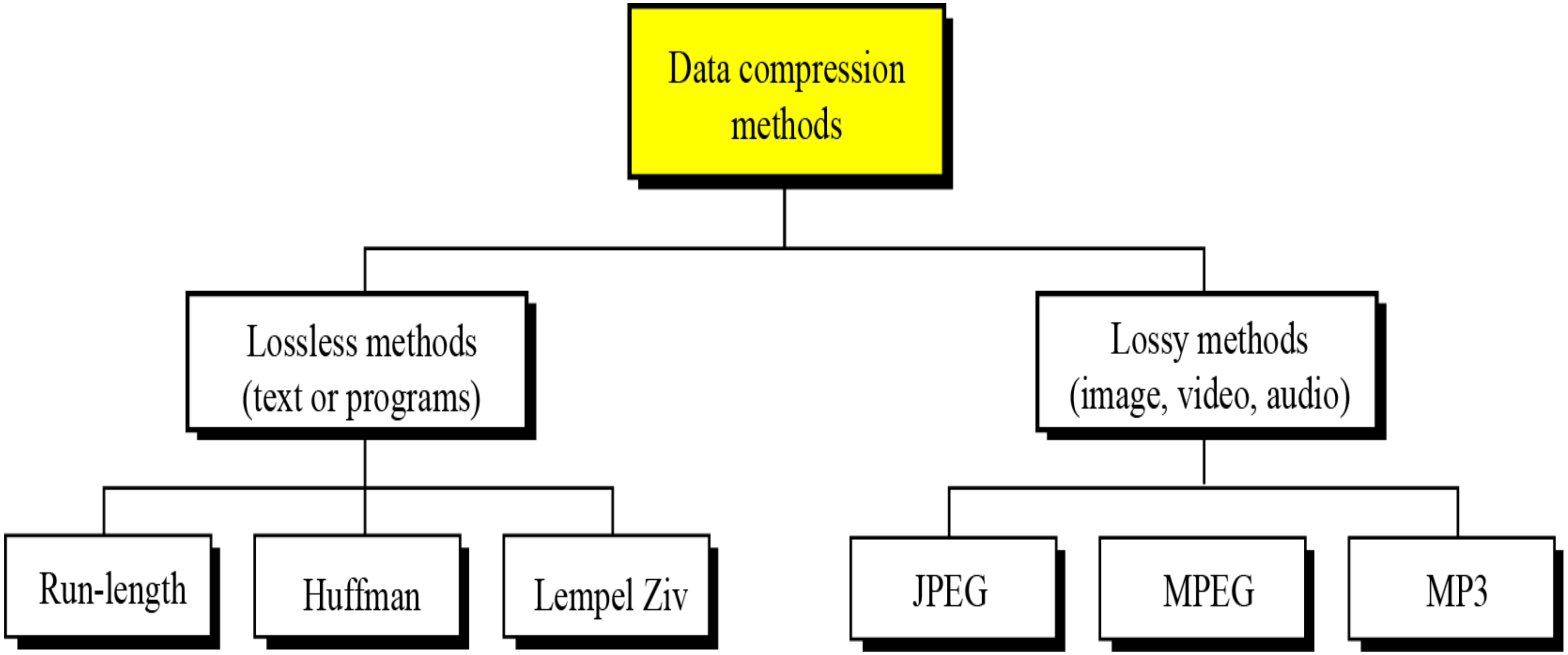


Compression

- Although many methods are used for this purpose, in general these methods can be divided into two broad categories:-

1) Lossless compression

2) Lossy Compression



MPEG-1

- The first standard to be finalized for video compression was MPEG-1for interactive video on CD’s and for digital audio broadcasting.

VCR	**640**	**480**	**24**	**FPS-25**	**368.64Mbps**	**MPEG-1 1.5 Mbps**

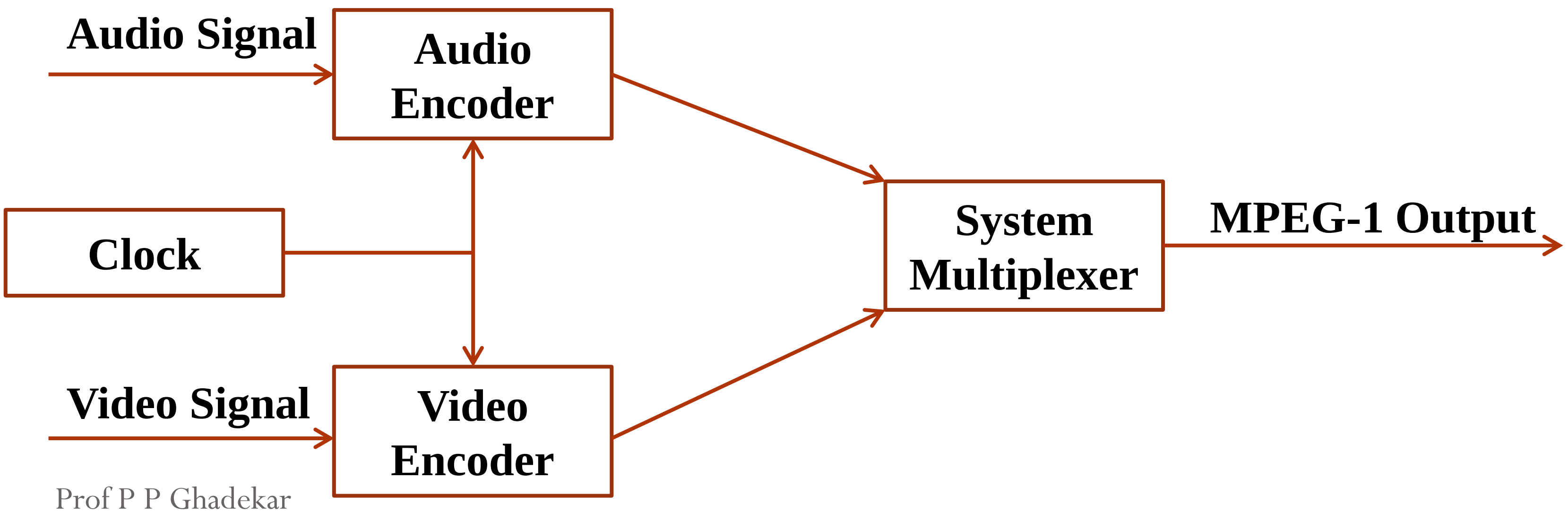
- It is likely to dominate the encoding of CD-ROM based movies.
- It can be transmitted over twisted-pair for modest distances.
- It support distance of 5KM or 15000 feet's.

MPEG-1

- Resolution-640 x 480 x 24. Frames per second-25. Total memory requirement in un compressed form is 368.64Mbps. By using MPEG-1 it required memory 1.5 Mbps, it causes the tremendous compression ratio.

MPEG-1

- MPEG-1 has three components; audio, video and system, which integrates the two.
- A 90KHz clock outputs the current time value (timestamps) to both the encoders and propagates all the way to the receiver.



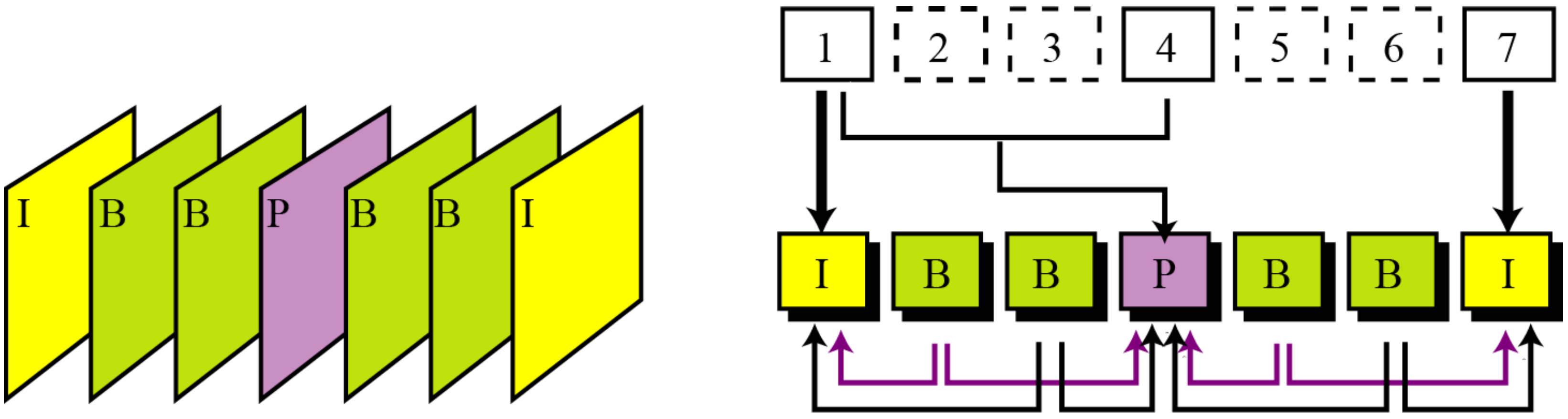
MPEG-1

- Encoding each frame separately with JPEG removes spatial redundancy.
- This is used where is need to access each frame randomly, such as editing video productions leading to 8-10Mbps compressed bandwidth.
- Additional compression can be achieved by taking advantage of the fact that consecutive frames are often almost identical.

MPEG-1 Frame Type

- MPEG-1 output consists of four kinds of frames for motion compensation.
  - P(Predictive) frames:-Block by block difference with the preceding I or P frame.
  - B(Bidirectional) frames:-Differences with the preceding and the following I or P frame.
  - I(Intracode) frames:- Self contained JPEG encoded, appears periodically.
  - D(DC-coded) frames:- Block averages used for fast forward.

Basics To Video Compression



a. Frames

b. Frame construction

Basics To Video Compression

- In videos like news programs, there is very little change from frame to another.
- Video codecs can take advantage of this property by only storing the differences from previous reference frames rather than storing the entire frame.
- This is implemented by doing motion estimation and compensation.

I-Frame

- **I(Intracode)frame:-** Certain frames in a video are designated as key frames and these frames are not compressed. Such frames are called as "Intra frames" or simply "I-Frames".
- They may also be known as **I-pictures**, or **key frames** due to their somewhat similar function to the key frames used in animation.
  - It is an independent frame that is not related to any other frame.
  - They are present at regular intervals.
  - I-frames are independent of other frames and cannot be constructed from other frames.

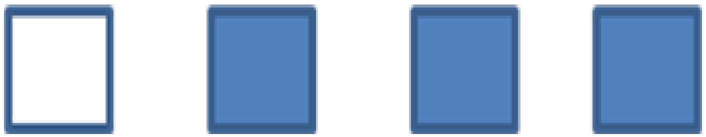


Figure: I-frame

P-Frame

- **P(Predictive)Frames:-** Frames that are stored as differences from a previous reference frame are called as “P-Frames”.
- P-frames exist to improve compression by exploiting the temporal (over time) redundancy in a video.
  - It is related to the preceding I-frame or P-frame.
  - Each P-frame contains only the changes from the preceding frame.
  - P-frames can be constructed only from previous I- or P-frames.

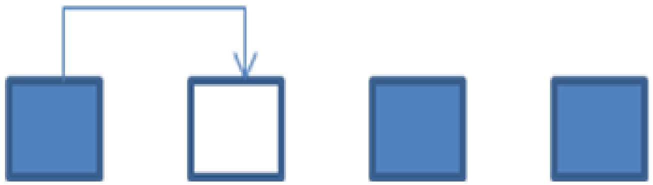


Figure : P-frame

B-Frame

- **B(Bidirectional)Frames:-** "B-Frames" are frames that are constructed by using both a previous frame and a future frame as a reference.
  - It is relative to the preceding and following I-frame or P-frame.
  - Each B-frame is relative to the past and the future.
  - A B-frame is never related to another B-frame.

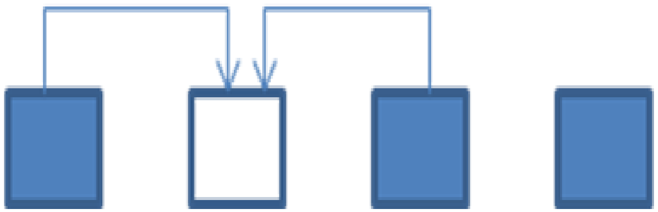
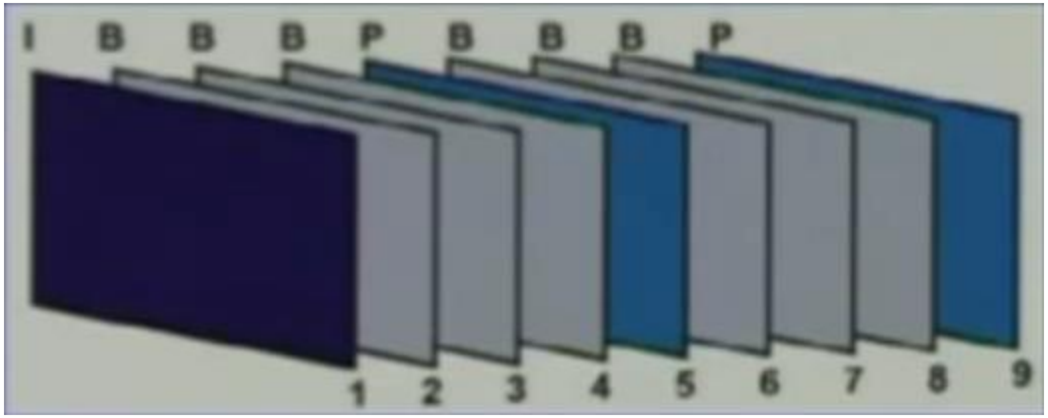


Figure: B-frame

D-Frame

- **D (DC coded)-Frames:-** Block averages used for fast forward.
- MPEG-1 has a unique frame type not found in later video standards.
- **D-frames** or **DC-pictures** are independent images (intra-frames) that have been encoded using DC transform coefficients only (AC coefficients are removed when encoding).
- D-frames are never referenced by I-, P- or B- frames.
- D-frames are only used for fast previews of video, for instance when seeking through a video at high speed.

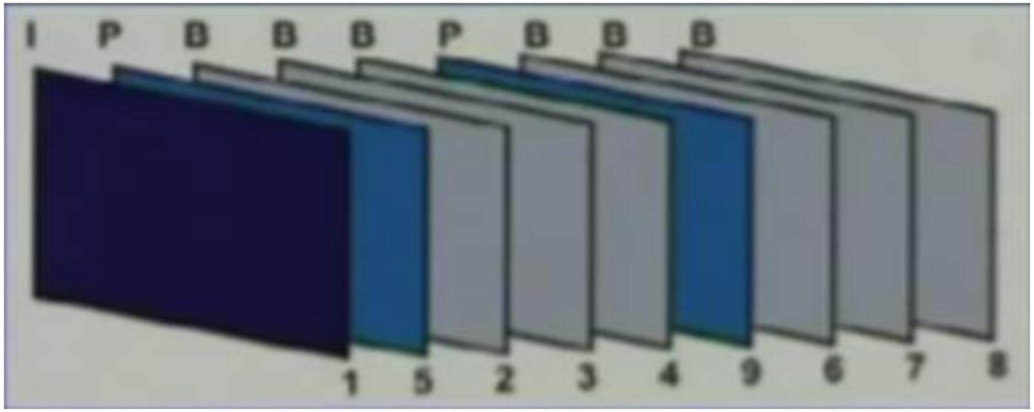
Coding and Display



Display Order

❖ Encoder can takes minutes or hours to do the encoding of Data , but decoder has to be fast.

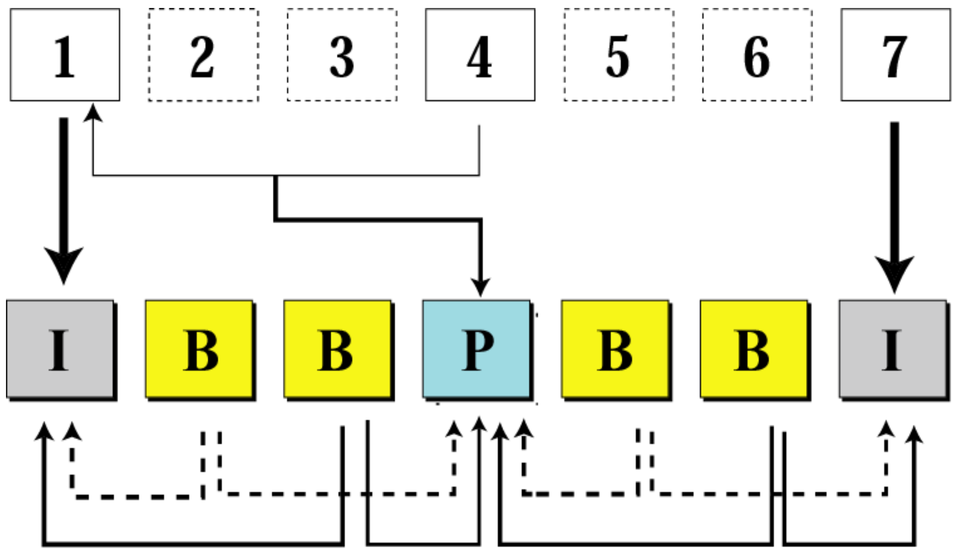
❖ Coding and Display order can be different



Coding Order

Frame construction

Frame construction can be better visualized by the figure below.



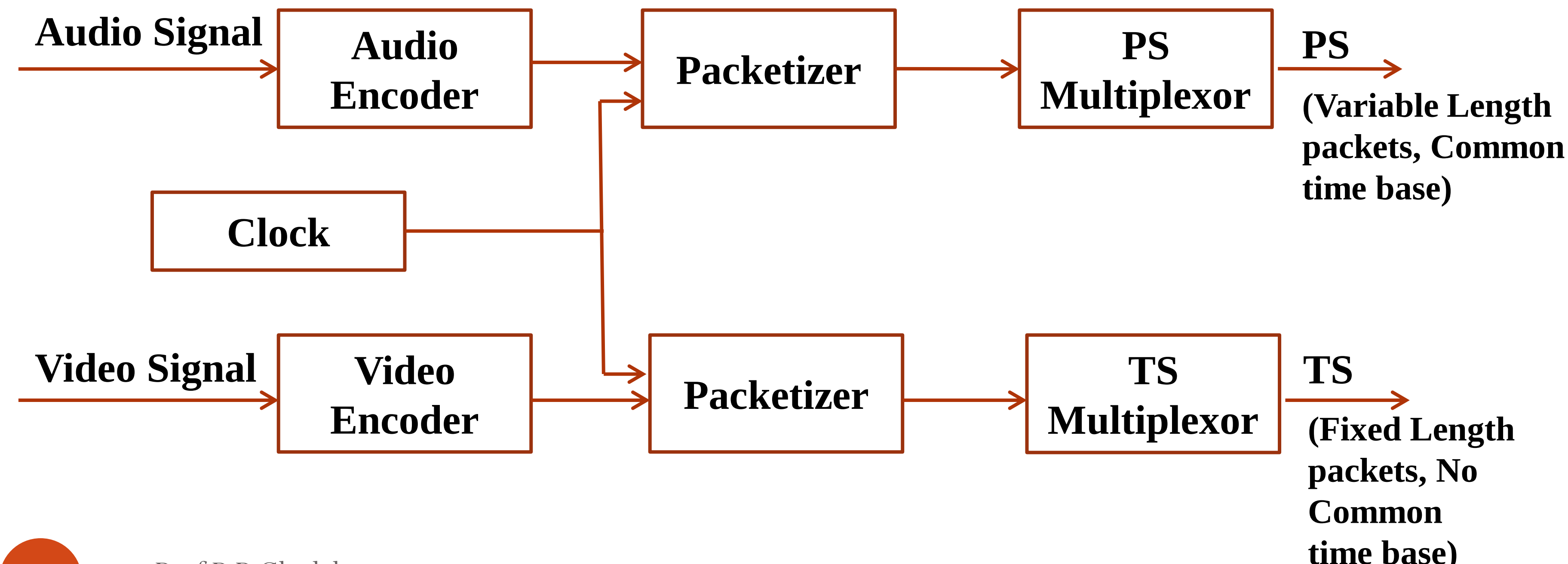
MPEG-2

- Although MPEG-2 is similar to MPEG-1, it was developed for digital television.
- Differences
- D-Frames are not supported.
- DCT is 10x10 instead of 8x8 to provide better quality.
- Support four resolutions.
- Supports five profiles(for different application)

HDTV	**1920**	**1080**	**24**	**60**	**2986 Mbps**	**MPEG-2 25-34 Mbps**
TV	**720**	**576**	**24**	**25**	**498 Mbps**	**MPEG-2 3-6 Mbps**

MPEG-2

- MPEG-2 has a more general way of multiplexing.
- Each streams are packetized with time stamps
- The output of each packetizer is a packetized Elementary System(PES) having 30 header fields. PS-Program Sequence



Compression transforms

- There are numbers of algorithms are available for reviling the redundancy(Spatial and Temporal) between pixels.
- The algorithms are…

1)Discrete Cosine Transform

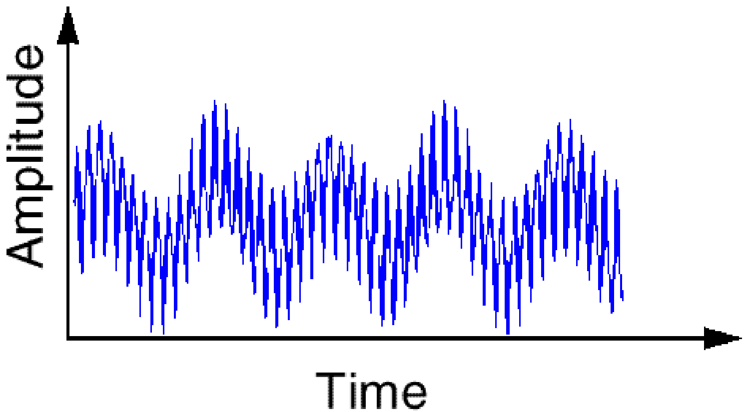
2)Discrete Fourier Transform

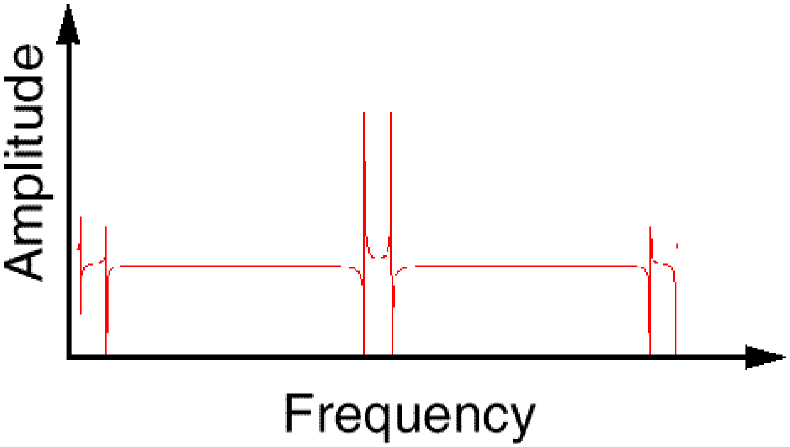
3)Fractal Coding

4)Wavelet Transform etc….

Fourier Analysis

- Breaks down a signal into constituent sinusoids of different frequencies





In other words: Transform the view of the signal from time-base to frequency-base.

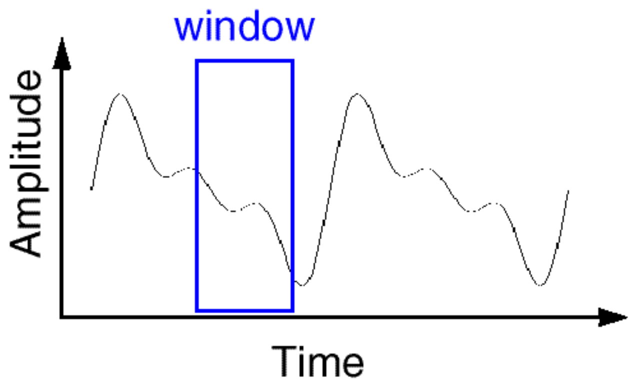
What's wrong with Fourier?

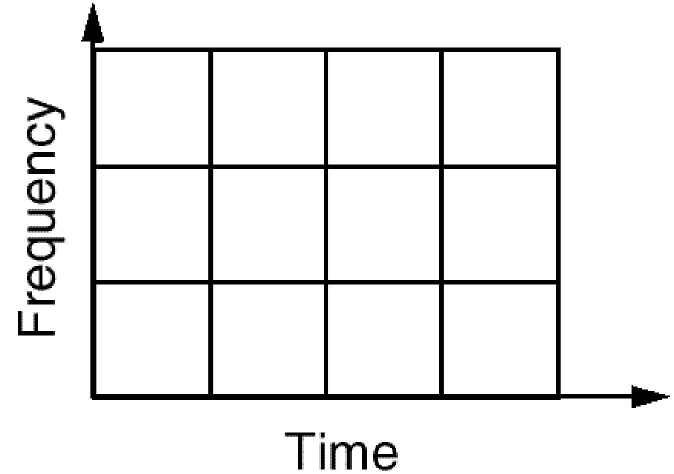
- By using Fourier Transform , we loose the time information : WHEN did a particular event take place ?
- FT can not locate drift, trends, abrupt changes, beginning and ends of events, etc.
- It uses complex numbers.

Short Time Fourier Analysis

- In order to analyze small section of a signal, **Denis Gabor** (1946), developed a technique, based on the FT and using *windowing* : STFT
- The STFT is a modified version of the Fourier transform.
- The Fourier transform is not an effective tool to analyse non-stationary signals.
- STFT and wavelet transforms are effective tools to analyse non-stationary signals.
- In STFT, the non-stationary signal is divided into small portions, which are assumed to be stationary.
- This is done using a window function of a chosen width, which is shifted and multiplied with the signal to obtain small stationary signals.

STFT





❑ Drawbacks of STFT

- ❖ Once a particular size time window is chosen, the window remains the same for all frequencies.
- ❖ To analyse the signal effectively, a more flexible approach is needed where the window size can vary in order to determine more accurately either the time or frequency information of the signal.
- ❖ This problem is known as the resolution problem.

STFT (or: Gabor Transform)

- A compromise between time-based and frequency-based views of a signal.
- Both time and frequency are represented in limited precision.
- The precision is determined by the size of the window.
- Once you choose a particular size for the time window - it will be the same for all frequencies.

Evolution of Wavelet Transform

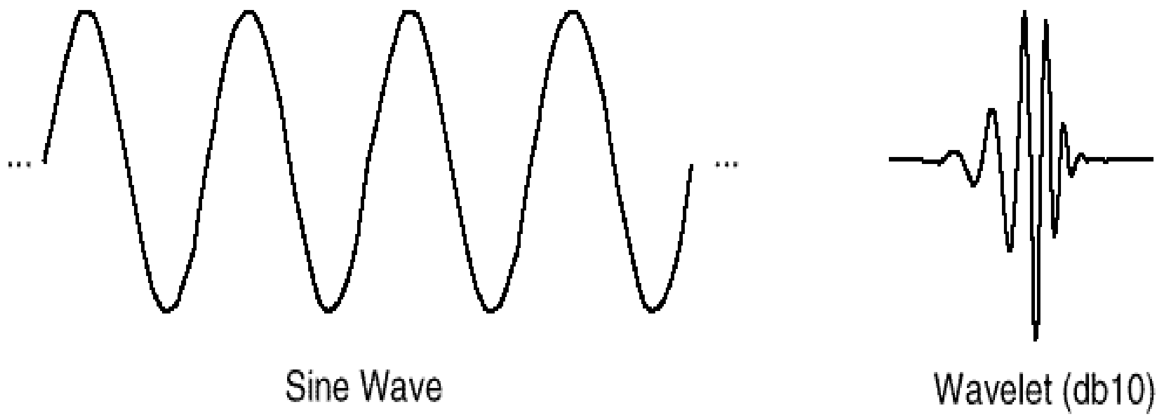
- Fourier transform is a powerful tool that has been available to signal analysis.
- It gives information regarding the frequency content of a signal.
- The problem with using Fourier transform is that frequency analysis cannot offer both good frequency and time resolution at the same time.
- A Fourier transform does not give information about the time at which a particular frequency has occurred in the signal.
- It is not effective tool to analyse a non-stationary signal.
- Wavelet transform is useful in image processing because it can simultaneously localise signals in time and scale.

Wavelet Transform

- A wavelet is a waveform of an effectively limited duration that has an average value of zero.
- The wavelet transform provides a time-frequency representation of the signal.
- The wavelet transform was developed to overcome the shortcoming of the Short time Fourier transform, which can be used to analyse non-stationary signals.
- Wavelet transform uses a multi-resolution technique by which different frequencies are analysed with different resolutions.
- The basic idea of the wavelet transform is to represent the signal to be analysed as superposition of wavelets.

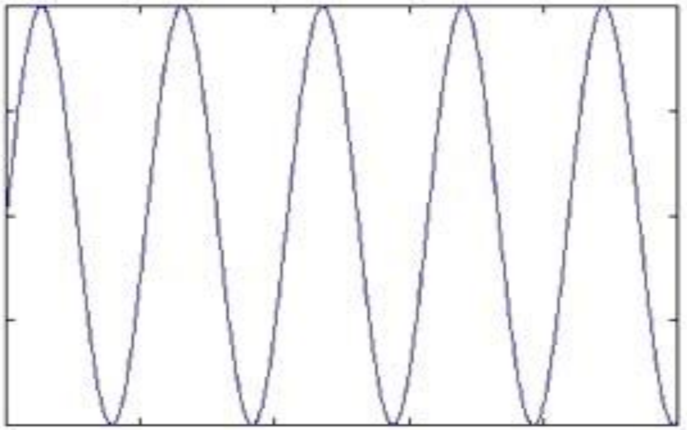
Wavelet Transform

❖ What is Wavelet?



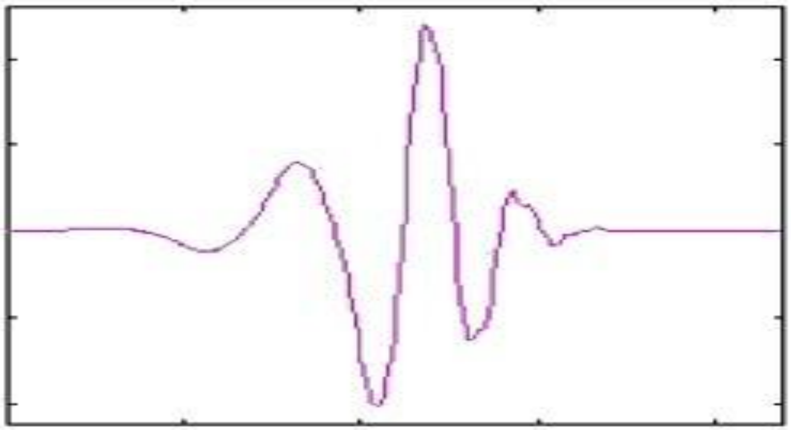
A wavelet is a waveform of effectively limited duration that has an average value of zero.

Wavelet



Sine wave

Smooth
Non-local (stretch out to infinity)



Wavelet

Irregular in shape Compactly supported (contained in finite domains)

Wavelet's Properties

- Short time localized waves with zero integral value.
- Possibility of time shifting.
- Flexibility.

Wavelet transform(Contd…)

- The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal.
- The Wavelet Transform provides a time-frequency representation of the signal.
- It was developed to overcome the short coming of the Short Time Fourier Transform (STFT), which can also be used to analyze non-stationary signals.
- STFT gives a constant resolution at all frequencies, the Wavelet Transform uses multi-resolution technique by which different frequencies are analyzed with different resolutions.
- Wavelet Transform categories into two different forms:-

  1)Continuous wavelet Transform

  2)Discrete wavelet Transform

The Continuous Wavelet Transform

- The *CWT* Is the sum over all time of the signal, multiplied by scaled and shifted versions of the wavelet function.

- The Wavelet functions are given by…

$$\boxed{\Psi_{a,b}(x) = \frac{1}{\sqrt{a}}\,\Psi\left(\frac{x-b}{a}\right)}$$

a-Scale Coefficient
b-Shift Coefficient

$$\boxed{\Psi_{a,b_x,b_y}(x,y) = \frac{1}{|a|}\,\Psi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right)}$$

2D Wavelet function

CWT(contd…)

- The Continuous Wavelet Transform (CWT) is provided by equation-

$$X_{WT}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \cdot \psi^{*}\left(\frac{t-\tau}{s}\right) dt$$

x(t) is the signal to be analyzed. ψ(t) is the mother wavelet or the basis function. All the wavelet functions used in the transformation are derived from the mother wavelet through translation (shifting) and scaling (dilation or compression).

CWT(Contd…)

- The mother wavelet used to generate all the basis functions is designed based on some desired characteristics associated with that function.
- The translation parameter t relates to the location of the wavelet function as it is shifted through the signal. Thus, it corresponds to the time information in the Wavelet Transform.
- The scale parameter *s is defined as |1/frequency| and corresponds to frequency information. Scaling either dilates (expands) or compresses a signal.*

Wavelet Properties

- A function $\Psi(X)$ can be called Wavelet if it posses the following Properties.

1. The function integrates to zero, or equivalently, its Fourier transform denoted as $\Psi(w)$ is zero at the origin.

$$\int_{-\infty}^{\infty} \Psi(X)dx = 0$$

   This implies $\Psi(w)|_{w=0}$ =0 in the frequency domain.

2. It is square integrable, or equivalently, has finite energy.

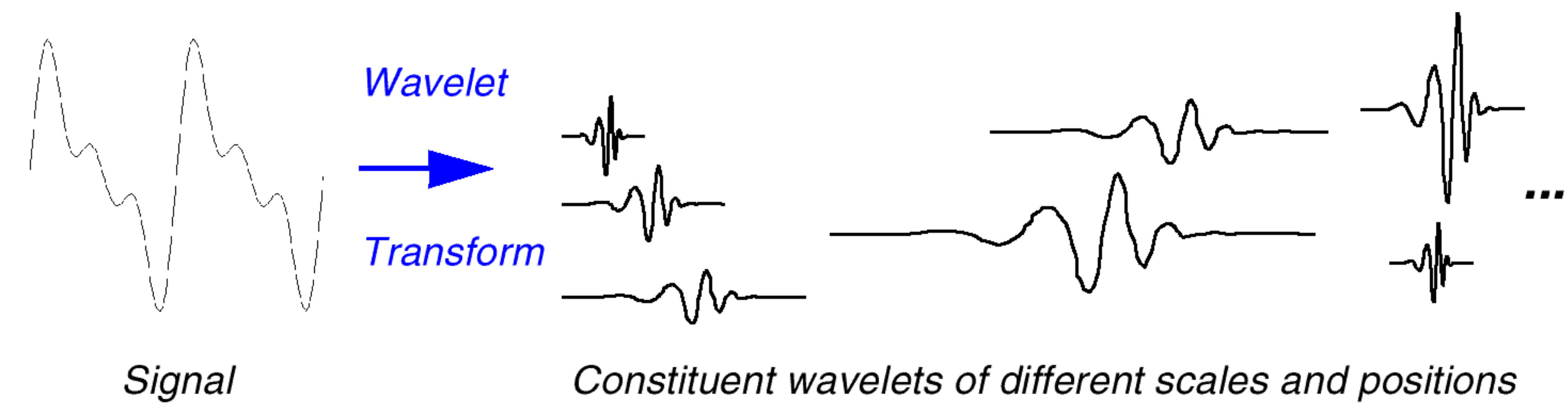
$$\int_{-\infty}^{\infty} | \Psi(X) |^2 \, dx < \infty$$

3. The Fourier transform $\Psi(X)$ must satisfy the admissibility condition given by

$$C_\Psi = \int_{-\infty}^{\infty} \frac{|\Psi(w)|^2}{|w|} dw < \infty$$

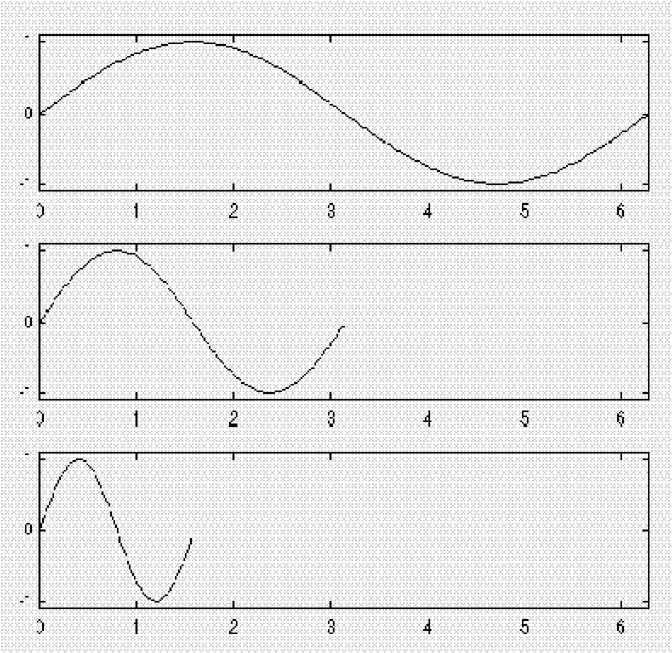
CWT(Contd…)

- The result of the CWT are Wavelet coefficients .
- Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelet of the original signal:



Scaling

- Wavelet analysis produces a Time scale view of the signal.
- *Scaling* means stretching or compressing of the signal.
- Scale factor (a) for sine waves:



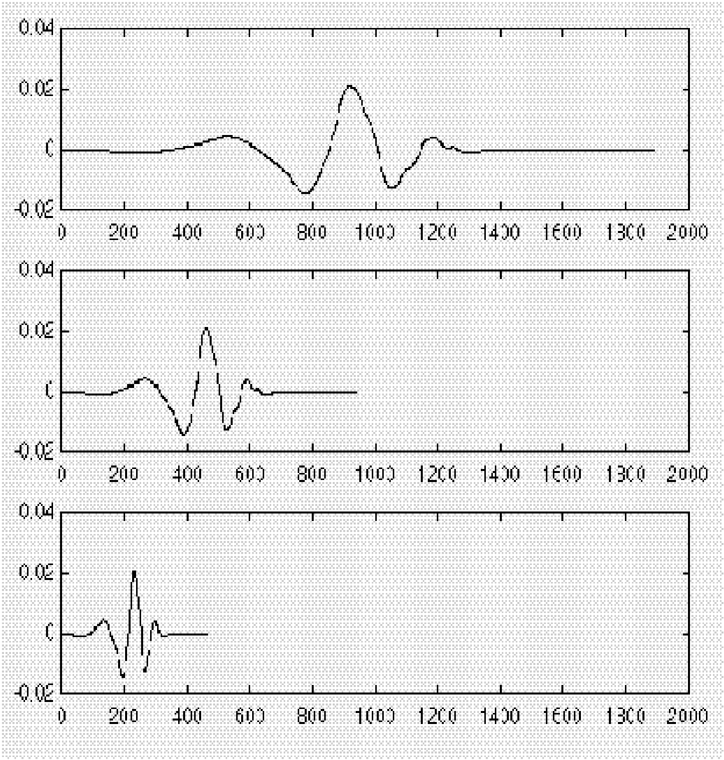
$$f(t) = \sin(t) \;\; ; \;\; a = 1$$

$$f(t) = \sin(2t) \;\; ; \;\; a = \tfrac{1}{2}$$

$$f(t) = \sin(4t) \;\; ; \;\; a = \tfrac{1}{4}$$

Scaling(Contd…)

- Scale factor works exactly the same with wavelets:



$$f(t) = \Psi(t) \;;\; a = 1$$

$$f(t) = \Psi(2t) \;;\; a = \tfrac{1}{2}$$

$$f(t) = \Psi(4t) \;;\; a = \tfrac{1}{4}$$

Have A Close Look…

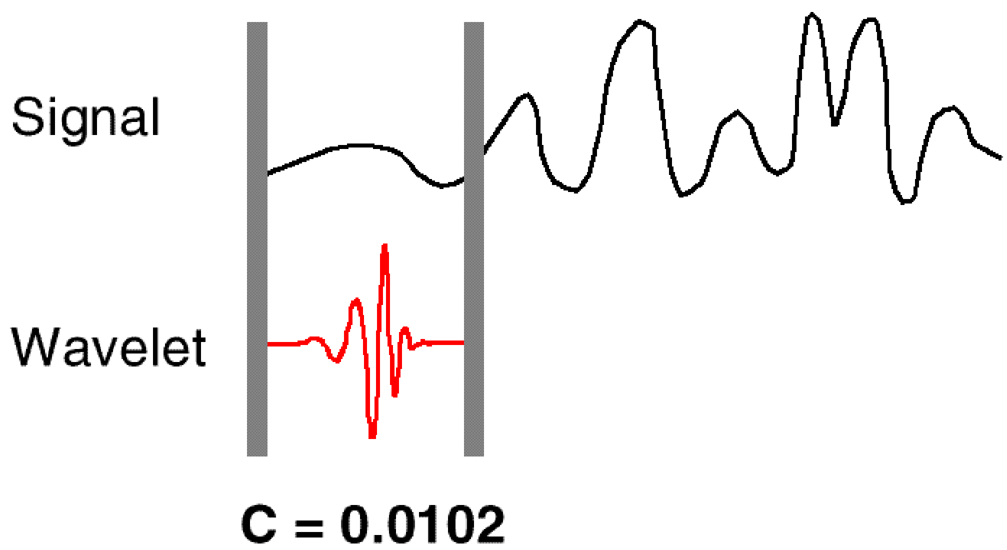
- Reminder:-The *CWT* Is the sum over all time of the signal, multiplied by scaled and shifted versions of the wavelet function.

<u>Step 1:</u>

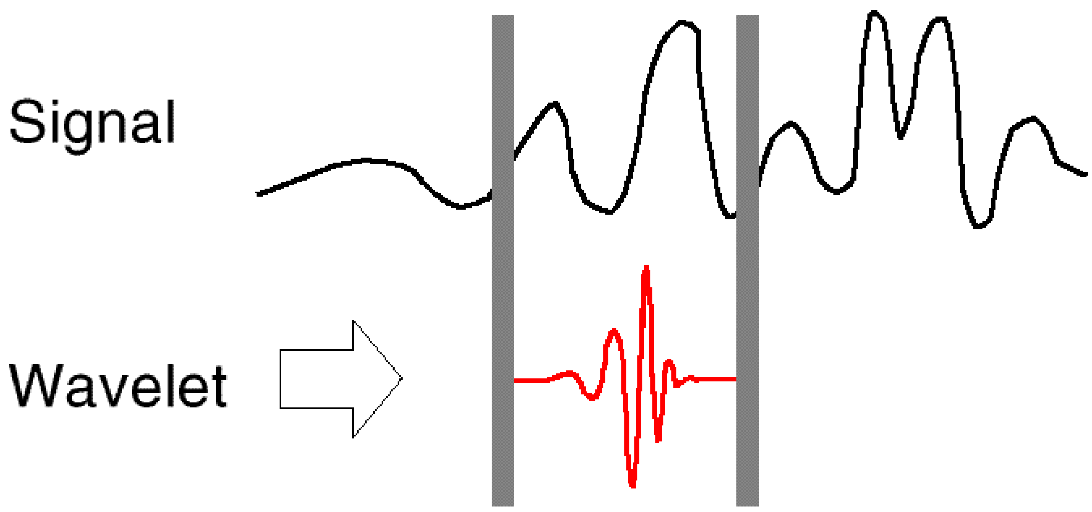
Take a Wavelet and compare it to a section at the start of the original signal

Step 2:

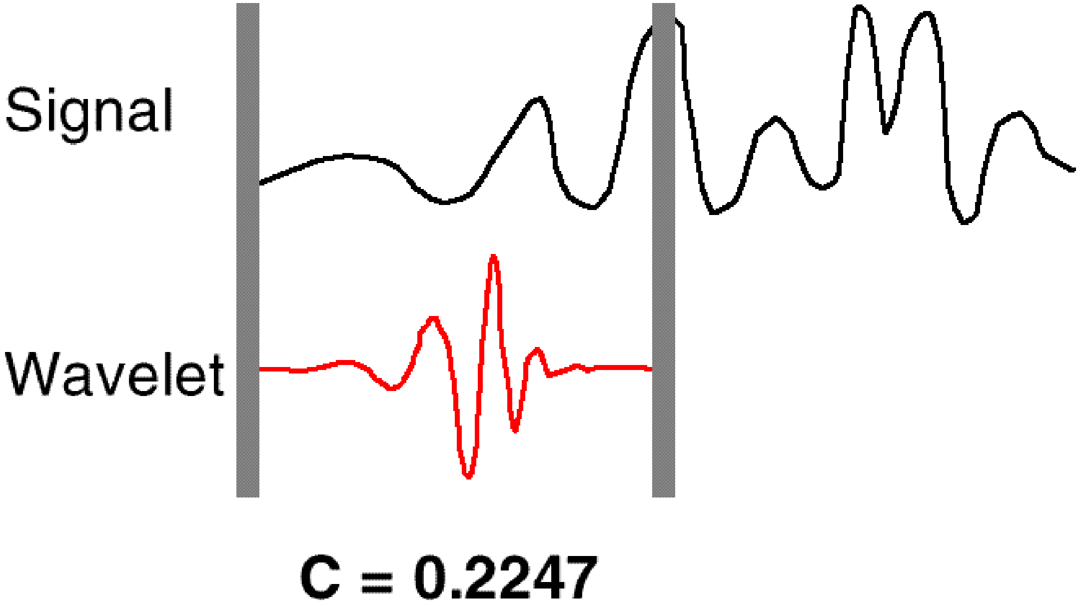
Calculate a number, C, that represents how closely correlated the wavelet is with this section of the signal. The higher C is, the more the similarity.



- Step 3: Shift the wavelet to the right and repeat steps 1-2 until you've covered the whole signal



- Step 4: Scale (stretch) the wavelet and repeat steps 1-3



STFT & DWT

1. The Fourier transforms does not computes negative frequencies.

2. In DWT the width of the window is changed as the transform is computed for every single spectral component, which is probably the most significant characteristic of the wavelet transform.

Desirable Properties of Wavelets

- Regularity
- Orthogonal and Bi-orthogonal Wavelet.
- Smoothness

Regularity

- The regularity of the wavelet-basis function is an important property of the wavelet transform.
- Due to this property, the wavelet transform can be local in both time and frequency domains.
- The regularity of the scaling function is determined by the decay of its Fourier transform $\Phi(w)$.

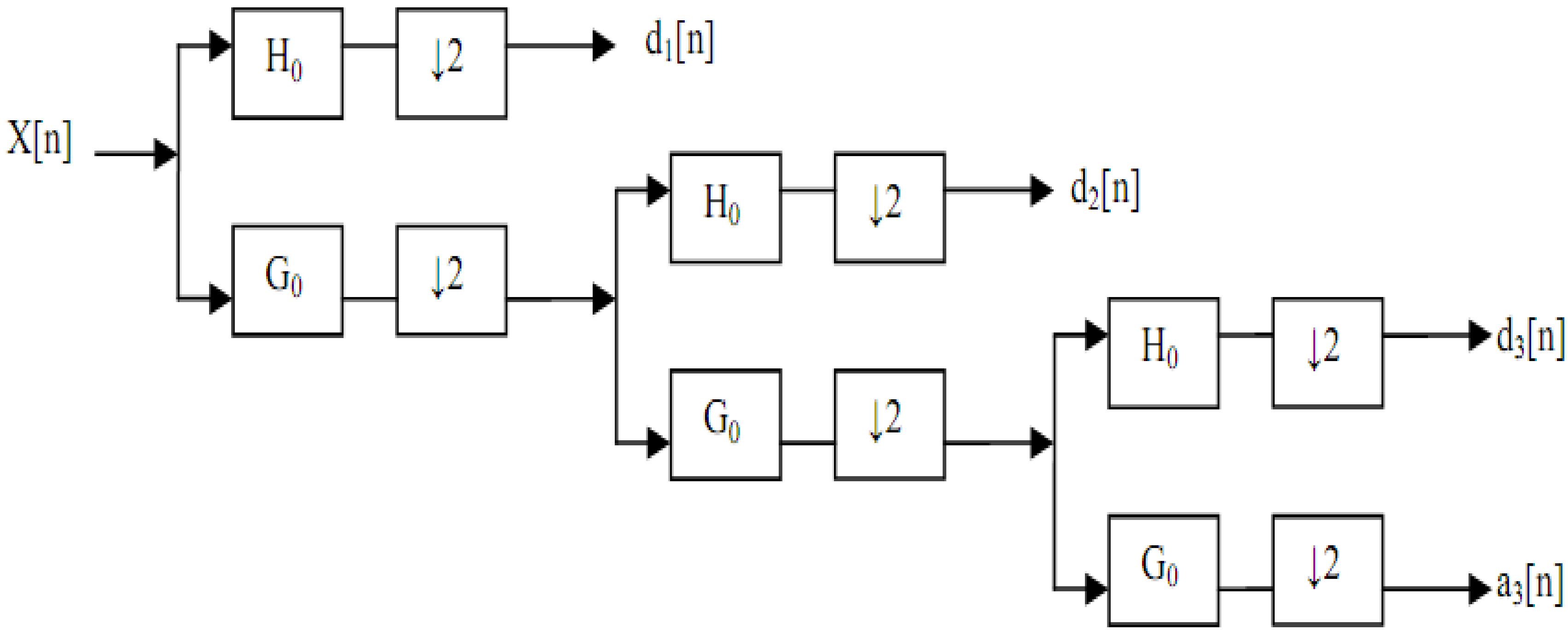
Discrete Wavelet Transform(DWT)

- The Wavelet Series is just a sampled version of CWT and its computation may consume significant amount of time and resources, depending on the resolution required.
- The Discrete Wavelet Transform (DWT), which is based on sub-band coding is found to yield a fast computation of Wavelet Transform.
- It is easy to implement and reduces the computation time and resources required.

DWT Decomposition

- The DWT is computed by successive lowpass and highpass filtering of the discrete time-domain signal.
- This is called the Mallat algorithm or Mallat-tree decomposition.
- Its significance is in the manner it connects the continuous time multiresolution to discrete-time filters.

DWT Decomposition(Contd…)



Three-level wavelet decomposition tree

DWT-Decomposition(Contd…)

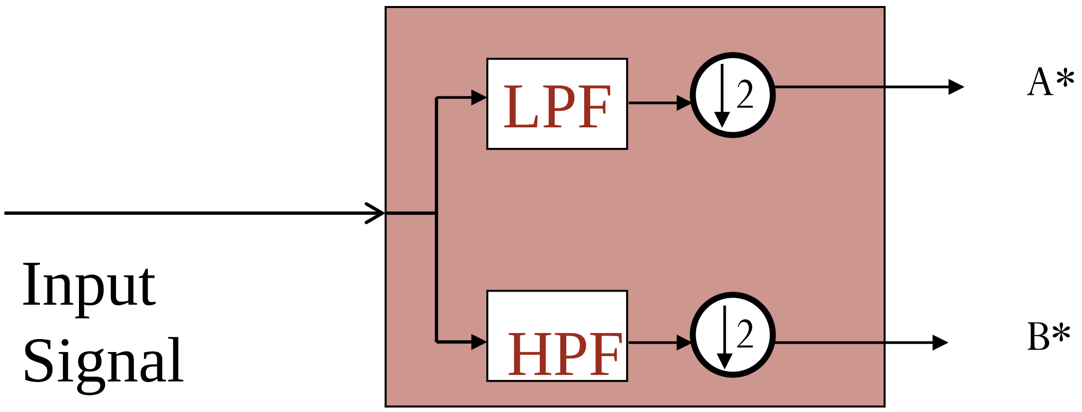
- In the figure, the signal is denoted by the sequence x[n], where n is an integer. The low pass filter is denoted by G0 while the high pass filter is denoted by H0.
- At each level, the high pass filter produces detail information, d[n], while the low pass filter associated with scaling function produces coarse approximations, a[n].

Decimation(Down Sample)

- The former process produces twice the data it began with: N input samples produce N approximations coefficients and N detail coefficients.
- To correct this, we *Down sample* (or: *Decimate*) the filter output by two, by simply throwing away every second coefficient.
- The amount of decimation depends on the ratio of the bandwidth of the filter output to the filter input.

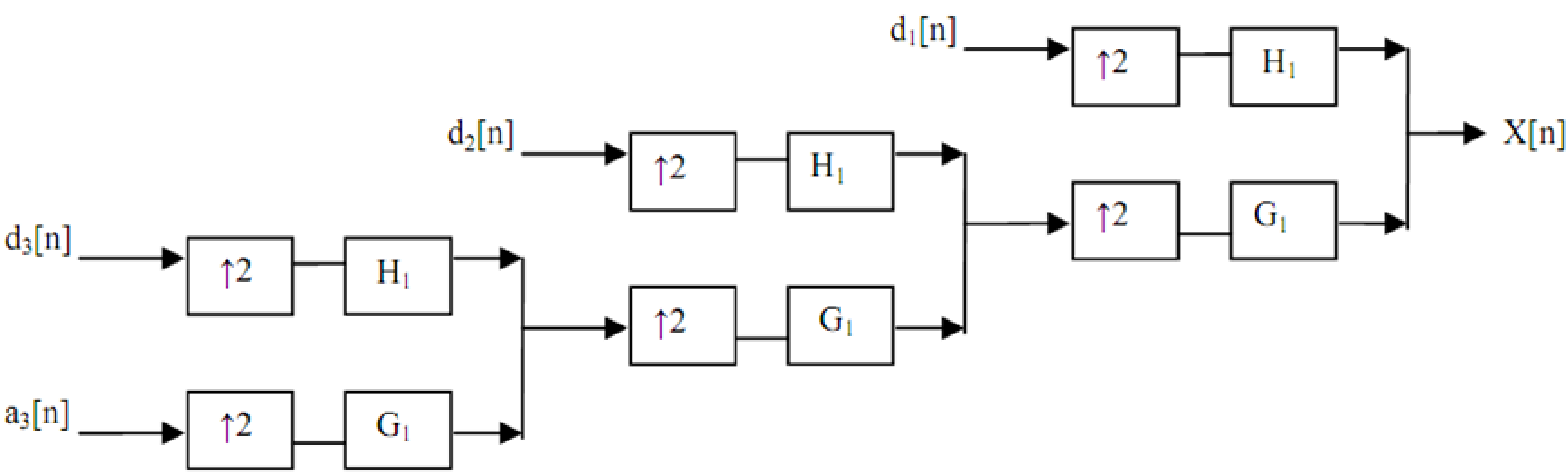
Decimation(Contd…)

- So, a complete one stage block looks like:



DWT Reconstruction (Interpolation)

- The DWT of the original signal is then obtained by concatenating all the coefficients, a[n] and d[n], starting from the last level of decomposition.
- The synthesis procedure involves expanding the signals in each branch by two which is termed expansion or interpolation.
- The interpolation is achieved by inserting zeros between successive samples.



Three-level wavelet reconstruction tree

Advantages of Wavelet based Image Compression

- Wavelets have non uniform frequency spectra which facilitate Multi scale analysis.
- The multi-resolution property of the wavelet transform can be used to exploit the fact that the response of the human eye is different to high and low frequency of an image.
- DWT can be applied to an entire image without imposing block structure as used by the DCT, there by reducing blocking artefact.

DWT implementation

- DWT implementation
- A Discrete Wavelet Transform (DWT) can be implemented through Filter bank scheme (Sub band Coding Scheme)or lifting scheme.

Applications of wavelets

- Some areas of application :

1. *Image processing*

- Increasing of quality image, image compression (wavelets are base of MPEG4)

2. *Signal processing :*

- Noise reduction, compression, coding, analysis of non stationary data

- *Other examples of wavelet applications are in astronomy, stock market, medicine, nuclear engineering, neurophysiology, music, optics etc.*

Sub-band Coding

- Sub-band coding is a procedure in which the input signal is subdivided into several frequency bands.
- Sub-band coding can be implemented through a filter bank.
- A filter bank is a collection of filters having either a common input or common output.
- When the filters have a common input, they form an analysis bank and when they share a common output, they form a synthesis bank.
- The basic idea in a filter bank is to participation a signal dyadically at the frequency domain.

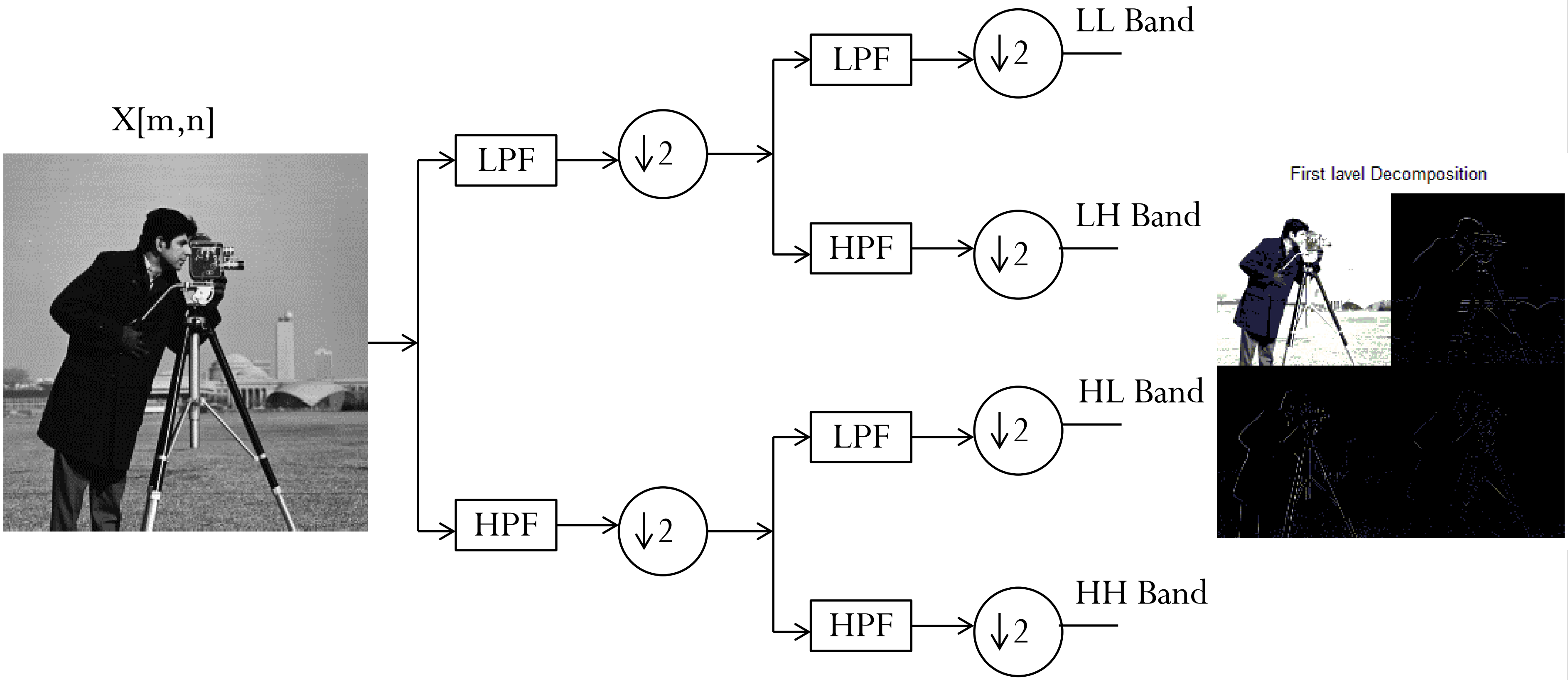
Sub-band Coding of 2-D Signal

- In the discrete wavelet transform, an image signal can be analysed by passing it through an analysis filter bank followed by decimation operation.
- The analysis filter bank consists of a low pass and high pass filter at each decomposition stage.
- When the signal passes through these filters, it splits into two bands.
- The low pass filter which corresponds to averaging operation, extract the coarse approximation information of the signal.
- The high pass, which corresponds to a differencing operation, extracts the details information of the signal.
- The output of the filtering operation is then decimated by two.

Sub-band Coding of 2-D Signal

- A two dimensional transform is accomplished by performing two separate one dimensional transforms.
- First, the image is filtered along the row and decimated by two.
- It is then followed by filtering the sub-image along the column and decimated by two.
- This operation splits the image into four bands, namely, LL, LH, HL, and HH respectively.

First Level of Decomposition of Cameraman Image

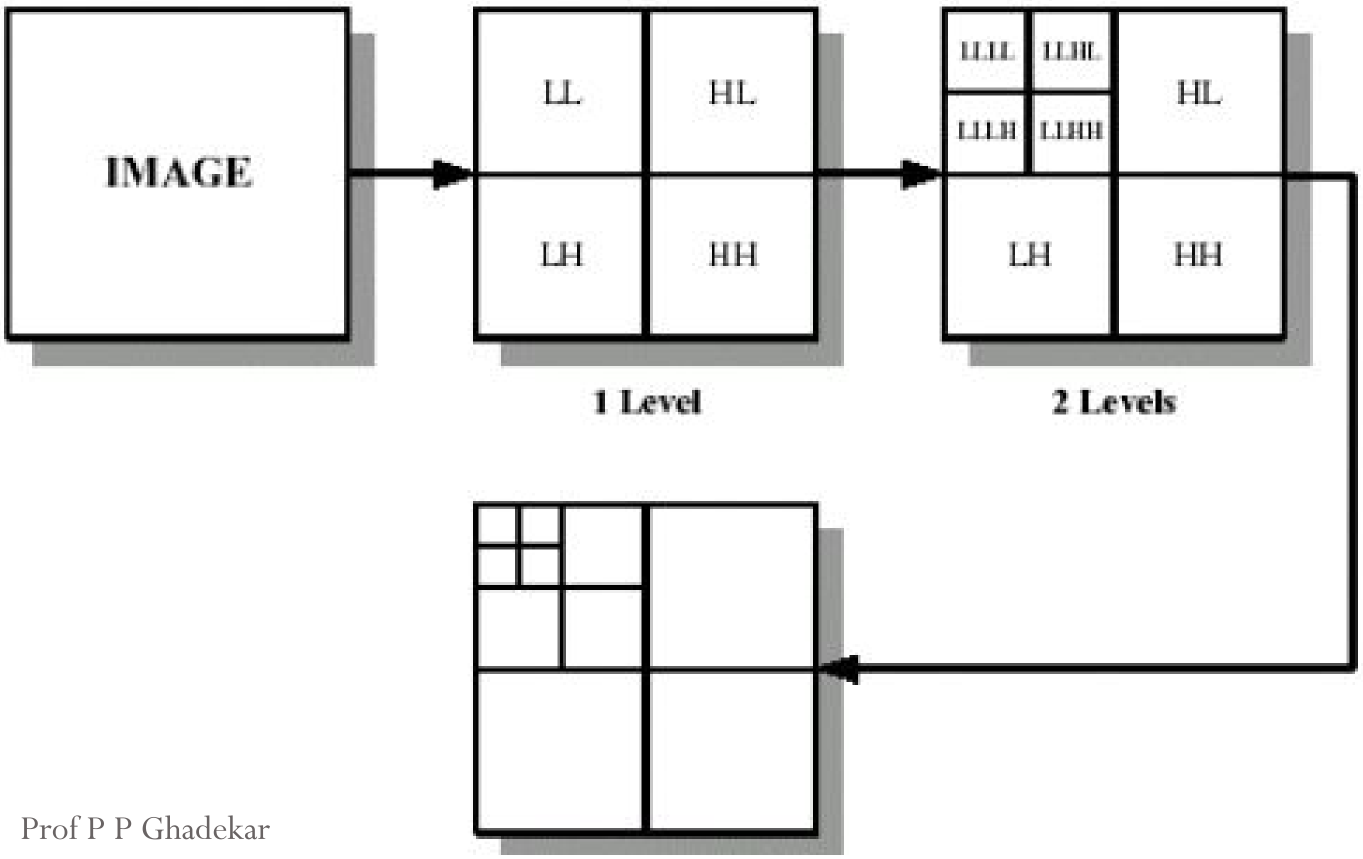


Sub-band Coding of 2-D Signal

- The LL piece comes from low pass filtering in both directions and it is the most like original picture and so is called the approximation.
- The remaining pieces are called detailed components. The HL comes from high pass filtering in the horizontal direction and low pass filtering in the vertical direction and so has the label HL. The visible detail in the sub-image, such as edges, have an overall vertical orientation
- LH comes from low pass filtering in the horizontal direction and high pass filtering in the vertical direction and so has the label LH, called horizontal details.
- The HH comes from high pass filtering in the horizontal and vertical direction and so has the label HH.

Second Level of Decomposition

Further Decomposition can be achieved by acting upon the LL sub-band successively and the resultant image is split into multiple bands as shown in figure.

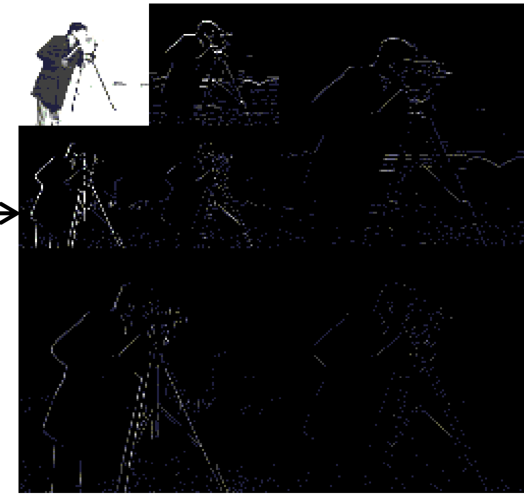


Discrete Wavelet Transform-Examples

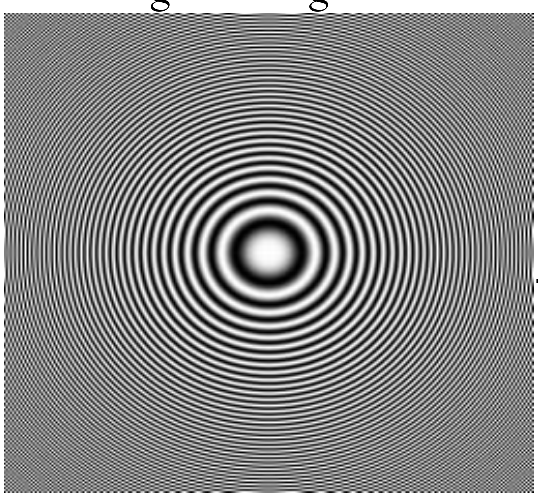
Original Image

First lavel Decomposition

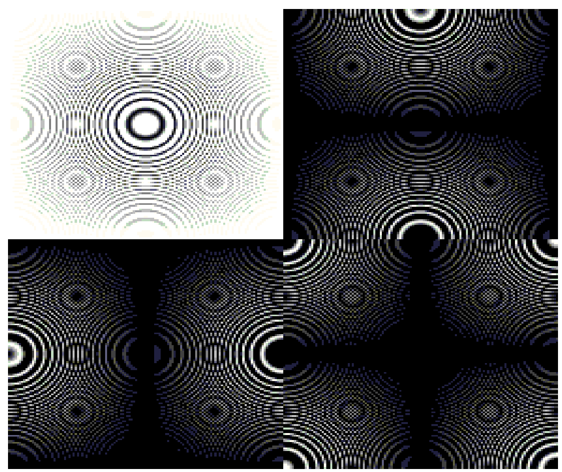
Second Level Decomposition



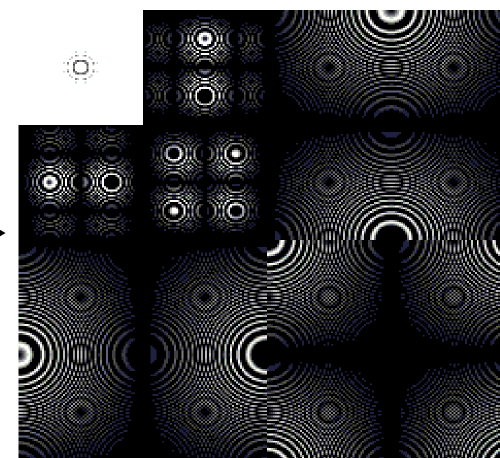
Original Image



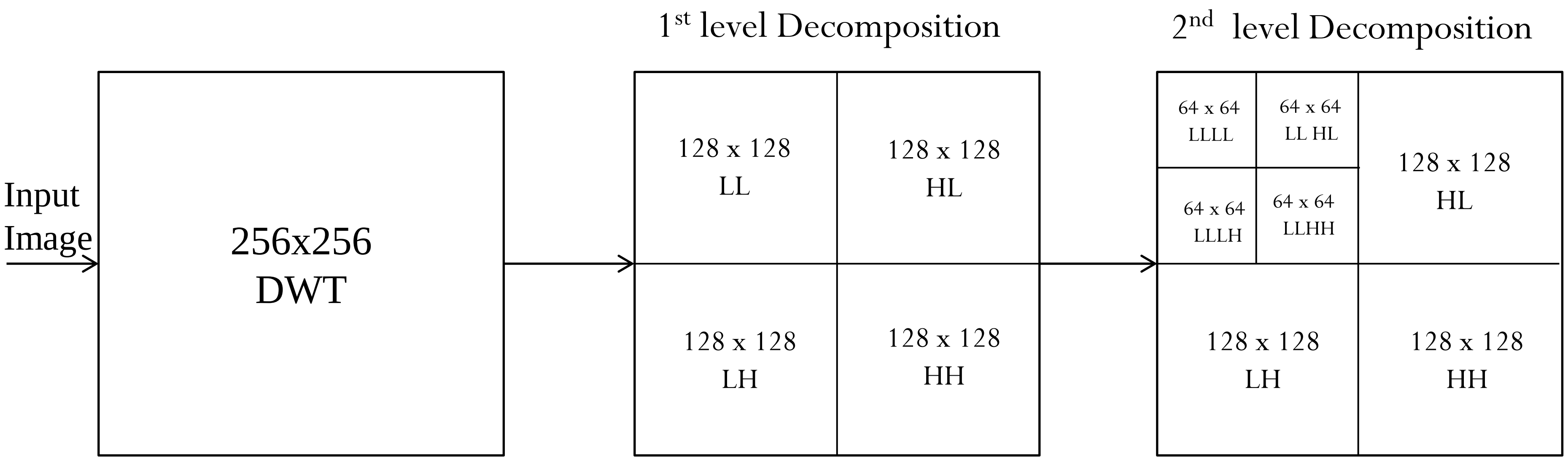
First lavel Decomposition



Second Level Decomposition

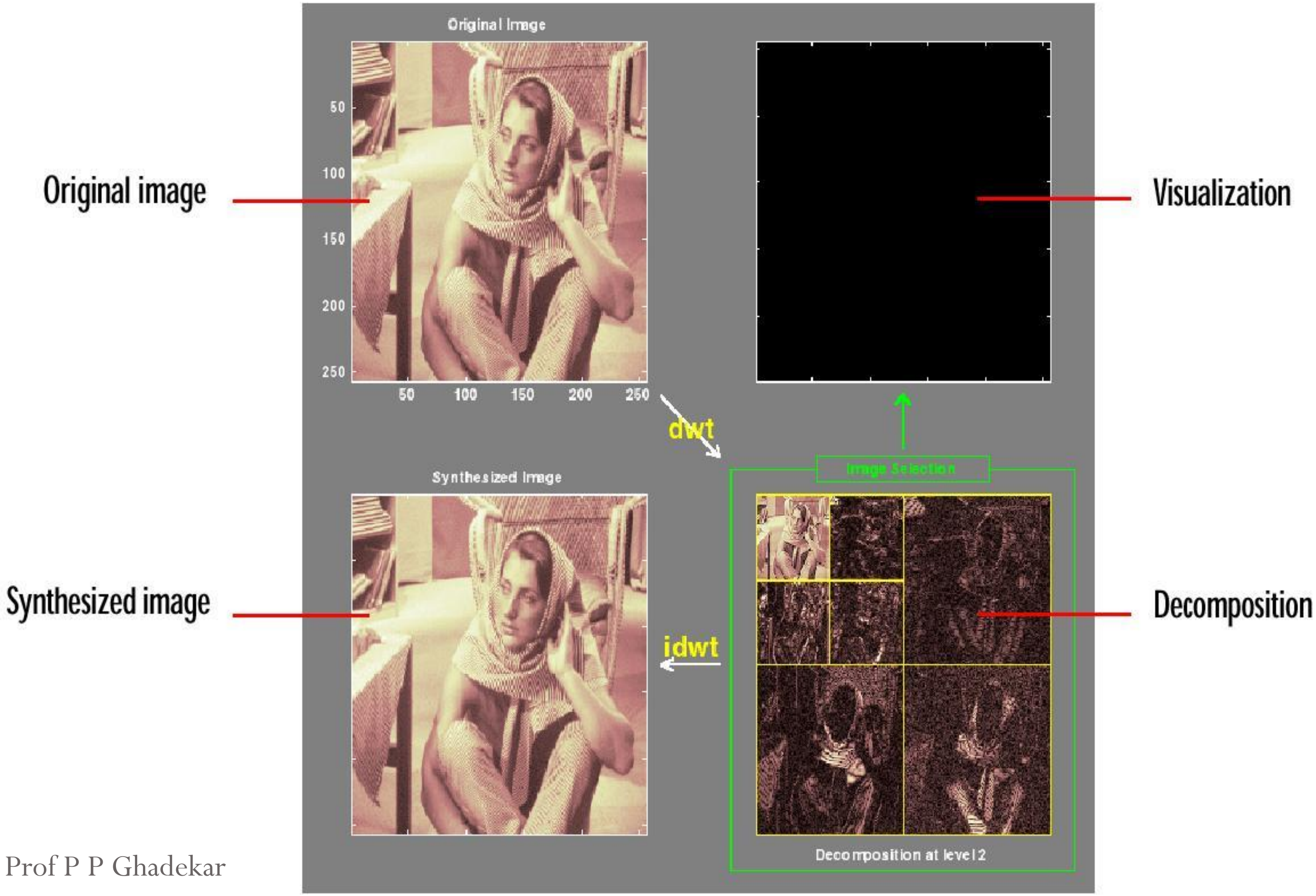


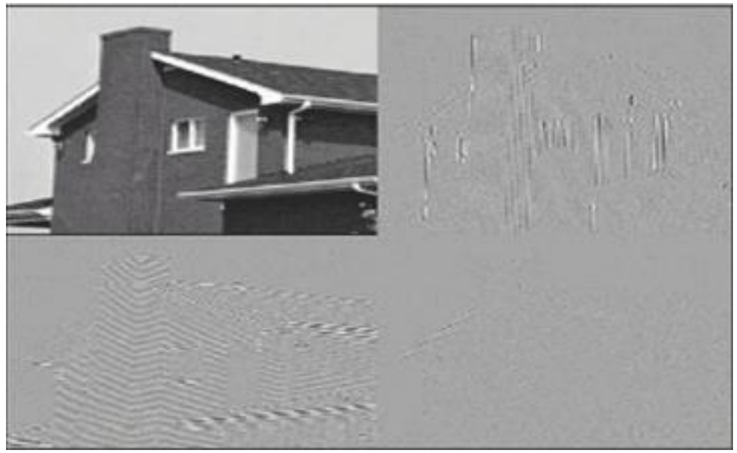
Second Level Decomposition of the Image

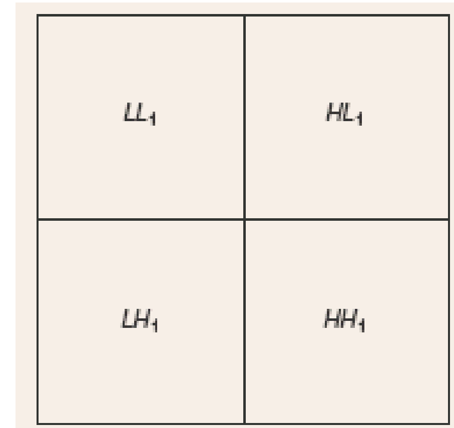


The size of the image at different levels of Decomposition

2-D Wavelet Transform via Separable Filters



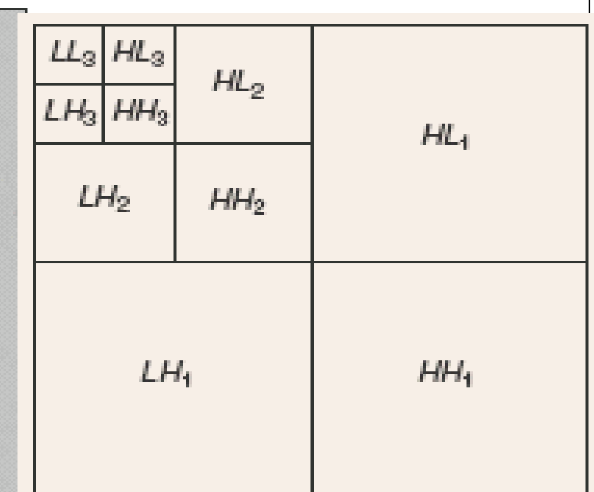




One Level 2D wavelet Transform

▲ 2. Original image used for demonstrating the 2-D wavelet transform.

Original Image



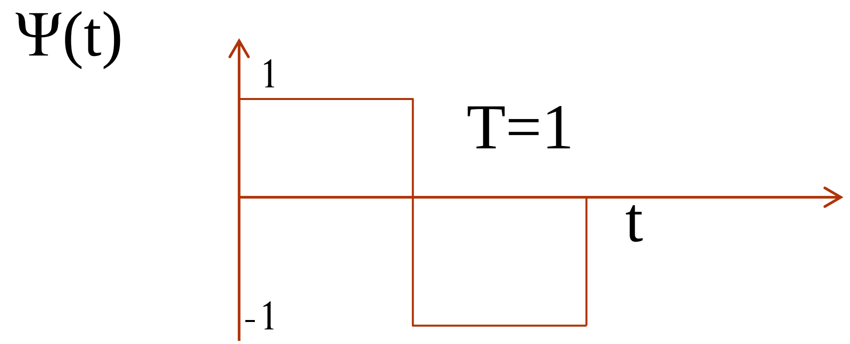
Examples of Wavelets

- Some of the most commonly used Wavelets. Most of them are continuous wavelets.
- Haar Wavelet
- Daubechies Wavelet
- Morelet Wavelet
- Mexican hat Wavelet

Haar Wavelet

- The Haar wavelet was developed by Haar in 1990.
- It is a bipolar step function.
- The expression of the Haar wavelet is given by

$$\Psi(t) = \begin{cases} 1 & \text{when } 0 < t < 1/2 \\ -1 & \text{when } 1/2 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$



- A Haar wavelet is the simplest type of wavelet. In discrete form, Haar wavelets are related to a mathematical operation called the Haar transform.
- The Haar transform serves as a prototype for all other wavelet transforms.

Haar Wavelet

- Haar transform decomposes a discrete signal into two subsignals of half its length. One subsignal is a running average or trend; the other subsignal is a running difference or fluctuation.
- The Haar wavelet is a real function, anti-symmetric with respect to $t=1/2$.
- The Haar wavelet is discontinuous in time.
- The Haar wavelet is localised in the time domain, but it has poor localization in the frequency domain.

Haar Wavelet Advantages

- The Haar wavelet transform has a number of advantages-

1)It is conceptually simple.

2)It is fast.

3)It is memory efficient, since it can be calculated without temporary array.

4)It is exactly reversible without the edge effects that are a problem with other wavelet transform.

Haar Wavelet (Limitations)

- The Haar window is only two elements wide. If a big change takes place from an even value to an odd value, the change will not be reflected in the high frequency coefficients.

Daubechies Wavelet

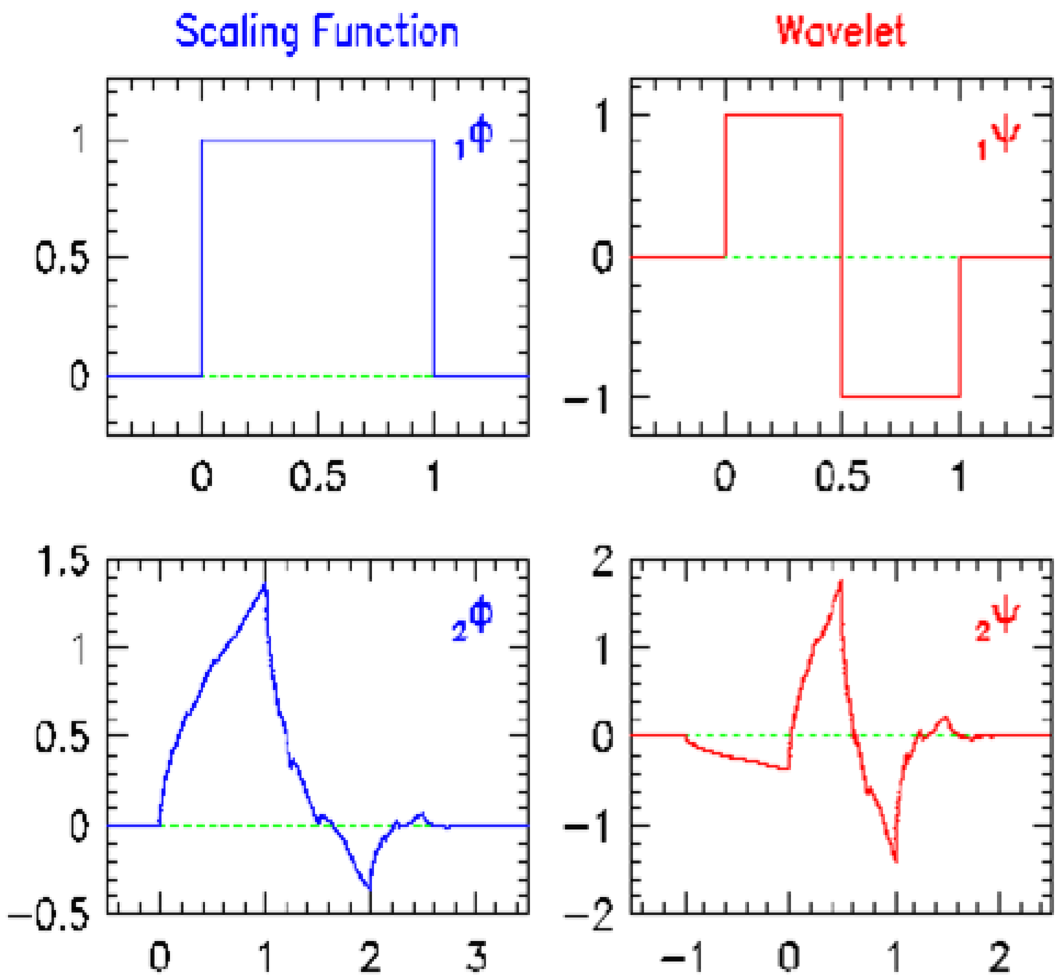
- The Daubechies wavelet transforms are defined in the same way as the Haar wavelet transform by computing the running averages and differences via scalar products with scaling signals.
- The only difference between them consists in how these scaling signals and wavelets are defined.
- The Daubechies wavelet is more complicated than the Haar wavelet.
- Daubechies wavelets are continuous; thus, they are more computationally expensive to use than the Haar wavelet

Daubechies Wavelet

- Daubechies wavelet has balanced frequency responses but non-linear phase responses.
- Daubechies wavelets use overlapping windows, so the high frequency coefficient spectrum reflects all high frequency changes.
- The Daubechies wavelet bases are a family of orthogonal, compactly supported scaling and wavelet functions that have maximum regularity for a given length of the support of the quadrature filters.

Wavelet functions examples

- Haar function
- Daubechies function



Haar Wavelet Eample

Let x and y be two data. The average and difference of these two numbers are given as follows:

$$s = \frac{x+y}{2}$$

$$d = y - x$$

Then, the reconstruction is possible using the following formula:

$$x = s - \frac{d}{2}$$

$$y = s + \frac{d}{2}$$

This can be illustrated in the following example.

Haar Wavelet Eample

Example 8.1 Consider the 1D array $[x,y]$ as [8, 6]. What are the approximation and detail coefficients?

Solution One can calculate the average s as $(8 + 6)/2 = 7$

The detail coefficients [1 2] in this process are obtained by taking the difference. Thus, the detail coefficient is given as −2.

$$6 - 8 = -2$$

Detail coefficients are important as they help retrieve the original array. This can be verified as follows:

$$x = s - \frac{d}{2} = 7 - \left(\frac{-2}{2}\right) = 8$$

$$y = s + \frac{d}{2} = 7 + \left(\frac{-2}{2}\right) = 6$$

Hence, this sequence is represented as follows:

$$[7 \; -2]$$

Performance Evaluation

- **Objective Quality Assessment**
  - Means Square Error (MSE)
  - Peak Signal to Noise Ratios (PSNR)
  - Compression Ratios (CR)
  - Similarity Structure Index (SSIM)
  - Execution Time (ET)
  - Time Complexity (TC)
- **Subjective Quality Assessment**

Objective Quality Assessment

- **Means Square Error(MSE)**
- MSE is calculated as

$$\mathrm{MSE} = \frac{1}{\mathrm{MN}} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(A_{i,j} - B_{i,j}\right)^2$$

Where, A is original frame of size M x N and B is synthesized frame of size M x N.

Objective Quality Assessment

- **Peak Signal to Noise Ratio (PSNR)**
- PSNR is evaluated as follows:

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{I}^2}{\mathrm{MSE}}$$

Where, 'I' is allowable image pixel intensity. 'MSE' is mean squared error.

Objective Quality Assessment

- **Compression Ratio (CR)**
- Compression ratio is the measure of reduction of information from original data. It is defined as:

$$CR = \frac{\text{Original Data} - \text{Compressed Data}}{\text{Original Data}} * 100$$

Objective Quality Assessment

- **Structural Similarity (SSIM) Index**
- During synthesis, structure of the frames gets distorted. This distortion can be analyzed by SSIM as:

$$\mathrm{SSIM}(A, B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}$$

Where,

μ_A, μ_B = mean intensities of original data A and synthesized data B;

σ_A, σ_B = standard deviation of original data A and synthesized data B;

Objective Quality Assessment

- C_1, C_2 = constant. Standard values of C_1 and C_2 are 0.01 and 0.03 respectively.
- Mean intensity can be calculated as below:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Standard deviation is mathematically calculated as:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

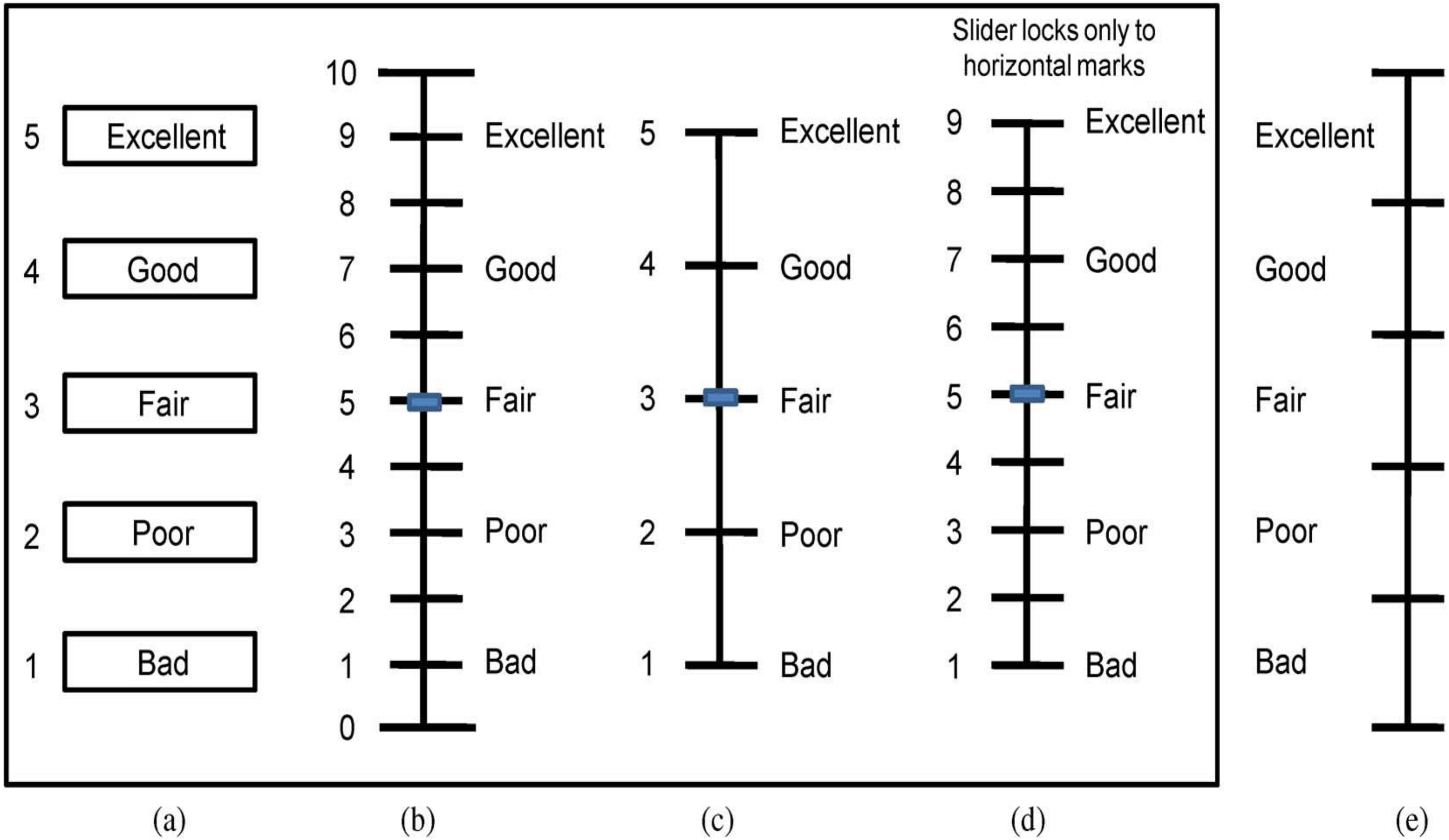
$$\sigma_{AB} = \frac{1}{N-1}\sum_{i=1}^{N}\left(A_i - \mu_A\right)(B_i - \mu_B)$$

- If synthesized frame is exactly similar to original frame, then SSIM is unity.

Subjective Quality Assessment

- **Five scale rating system**
- In this case, observers are requested to rate the quality of video on the scale of 5.
- ITU-R has recommended five scale rating system of degree of impairment.
- Table shows ITU-R Recommended 500 Quality Ratings on a scale from 1 to 5.

Rating	Impairment	Quality
5	Imperceptible	Excellent
4	Perceptible, not annoying	Good
3	Slightly annoying	Fair
2	Annoying	Poor
1	Very annoying	Bad



(a) 5-point discrete scale; (b) 11-point continuous scale; (c) 5-point continuous scale; (d) 9-point discrete scale. The standard ITU continuous scale is shown in (e) for information but was not used in our study.

Diolch
Kiitos
umesc
Sheun
Shnorhakalutiun
Kasih
Mamnoon
Todah
Te ekkür
Dekuju/Dekujeme
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
AČi
Ngiyabonga
Gamsahapnida
Dank
Takk
Cam
Xie
Gra
or
al
Kop
Salamat
Merci
Dhanyavaad
Waad
Dakujem
Dhanyavadaalu
Dankie
Go
Grazie
Faleminderit
Dhanyavad
Khopjai
Kruthagnathalu
Daw
ThankYou
kun
Arigatou
krap
Tack
Or
Dhonnobaad
Shukriya
ederim
Asante
Gracias
Nandree
Blagodariya
Gomapsupnida
Euxaristo
Kun
Hain
Dhan
Grazzi
raibh
Terima
Fyrir
Enkosi
Danke
dank
daa

Colour Model

By

Prof. (Dr.) Premanand P. Ghadekar

Outline

- On Completion students will be able to understand
  - Colour Image Processing
  - Primary and Secondary colours
  - Colour Characteristics.
  - Chromaticity diagram and its use.
  - Colour model
  - RGB colour model
  - HSI colour model
  - Conversion from one colour model to other.

Colour image processing

- Mainly there are two areas-
- Full colour Processing -It is used for full colour analysis. Images acquired with full colour sensor.

- Pseudo colour Processing- The process of adding colour to a gray scale image is called pseudo colour or false colour.

  It is used for Human Interpretation purpose.

Colour Spectrum

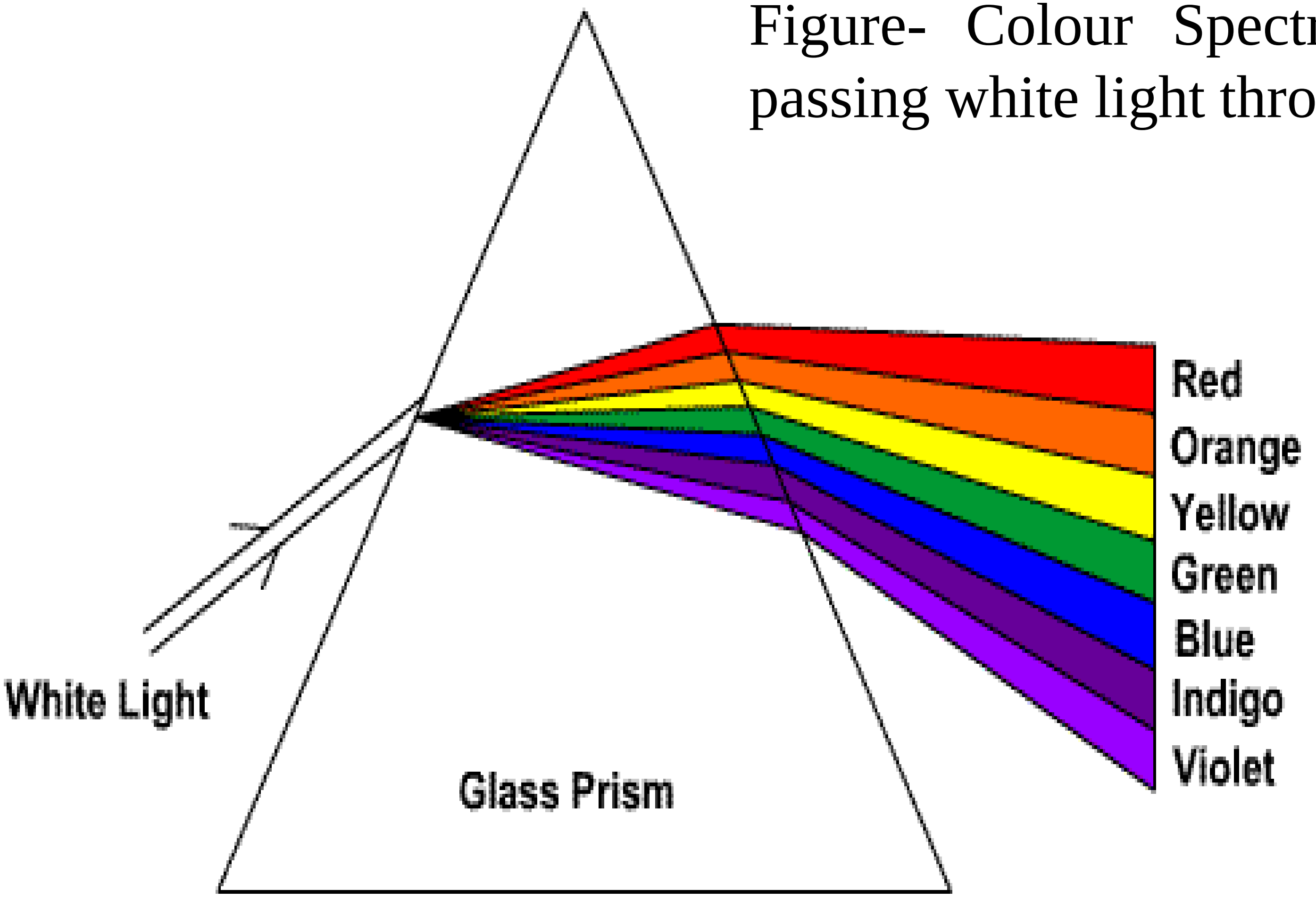


Figure- Colour Spectrum seen by passing white light through a Prism

ELECTROMAGNETIC SPECTRUM

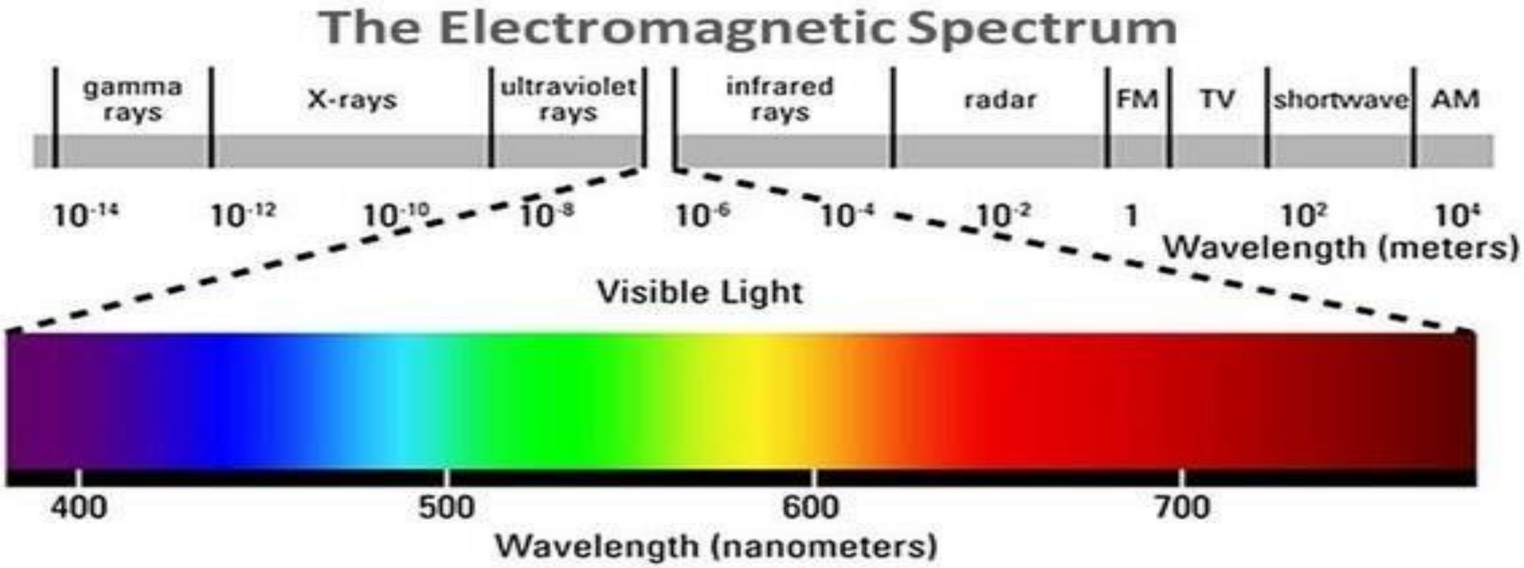


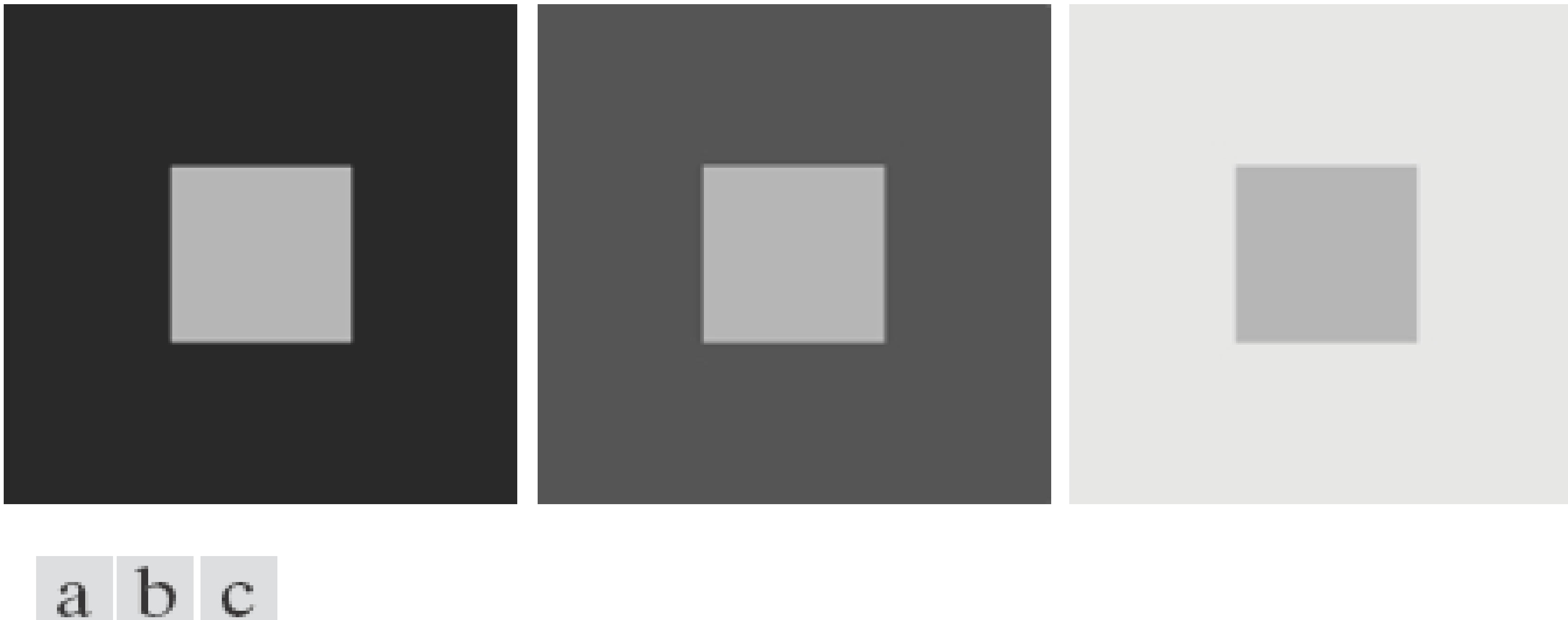
Figure -Wavelength comprising the visible range of the electromagnetic Spectrum

Quality of Colour

- Radiance- It is the total amount of energy i.e. come out from light source. It is measured in unit of watts.
- Luminance- It is amount of an energy perceived by an observer. It is measured in Lumens.
- Brightness – Perceived luminance of the surrounding.
- **Contrast -Contrast** is the difference in luminance and/or colour that makes an object (or its representation in an image or display) distinguishable.

Quality of Colour

- **Contrast Example**



- Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

Primary Colours

- There are three primary colours-Red, Green Blue
- Cones-More sensitive to colour components. 6 to 7 Millions.

- 65%-Red colour
- 33%-Green Colour
- 2%-Blue Colour

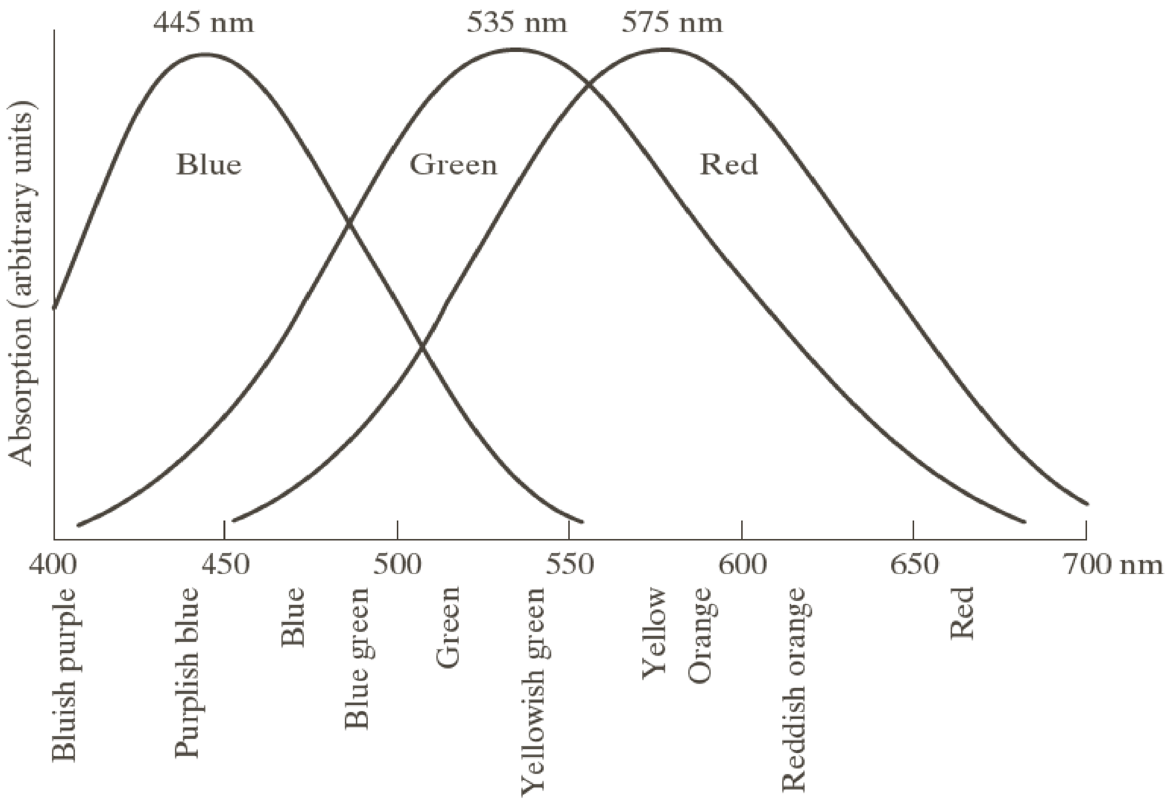
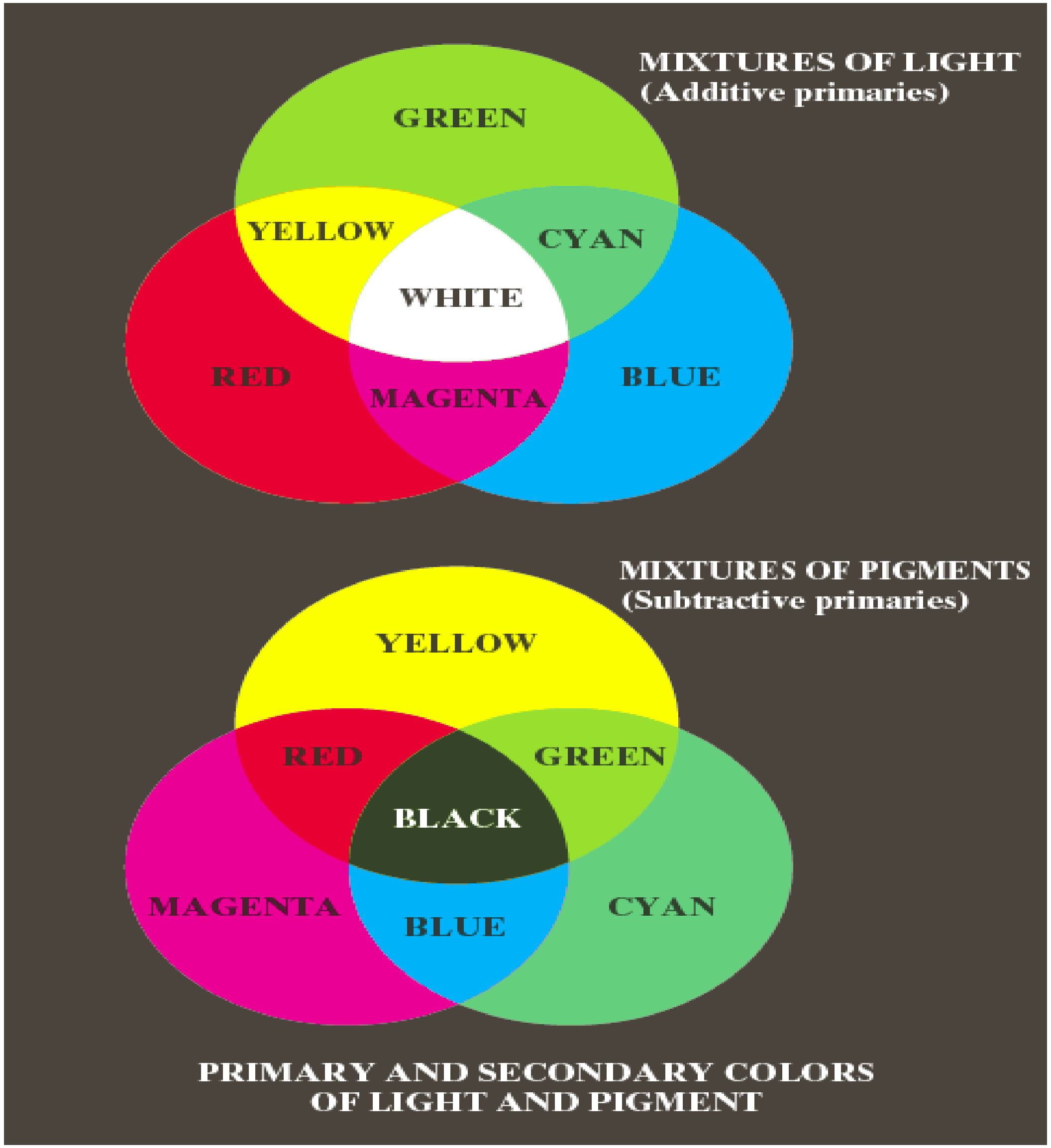


Figure- Absorption of light by the Red Green and Blue cones in the human eye as a Function of wavelength.

Primary and Secondary colors



Primary and Secondary Colours of light and Pigments

Subtractive Primaries-
Example:-Cyan=White-Red

PIGMENT

- A pigment is a material that changes the colour of reflected or transmitted light as the result of the wavelength selective absorption.
- Many materials selectively absorbs certain wavelength of light.
- A pigment must have a high tinting strength relative to the materials it colors.
- Pigments are used for coloring paints, ink, plastic, fabric, food, cosmetic and other materials.

Primary and secondary colours- Pigments

Red + Blue → Magenta

Green + Blue → Cyan

Red + Green → Yellow

$\bigg\}$ **Pigments**

Red

Green

Blue

$\bigg\}$ **Secondary Colours for Pigments**

Color Perception

There are two types of photoreceptors

- Rods
- Cones

Rods are responsible to recognize intensity/luminance of object.

Cones are responsible to recognize colour of object.

Sensitivity of the Human eye is greatest for green light

Colour Characteristic

Colour has three characteristics to specify visual information-

- Luminance-Intensity.
- Brightness-It is the perceived luminance of the surrounding.
- Hue-It is a colour attributes that describes a pure colour. It represents dominant colour as perceived by an observer.
- Saturation-It gives a measure of the degree to which a pure colour is diluted by white light. The degree of saturation is inversely proportional to the amount of white light added.

Example-Pink(white + red), Lavender (violet + white)

Colour models

- **RGB -Monitor**
- **CMY –Colour Printer**
- **CMYK –Colour Printer**
- These are Hardware Oriented colour model.

- **HSI-Hue Saturation and Intensity Colour model**
- It is application oriented/perception oriented.
- I- Gray scale information
- H & S- Chromatic information.

TRICHROMATIC COEFFICIENTS

- Hue and Saturation taken together are called Chromaticity (colour components).
- The amount of Red, Green, and Blue needed to form any particular colour called as-Tristimulus values(X,Y, Z).
- Its Tri-Chromatic coefficients are

$$x = X/(X+Y+Z)$$

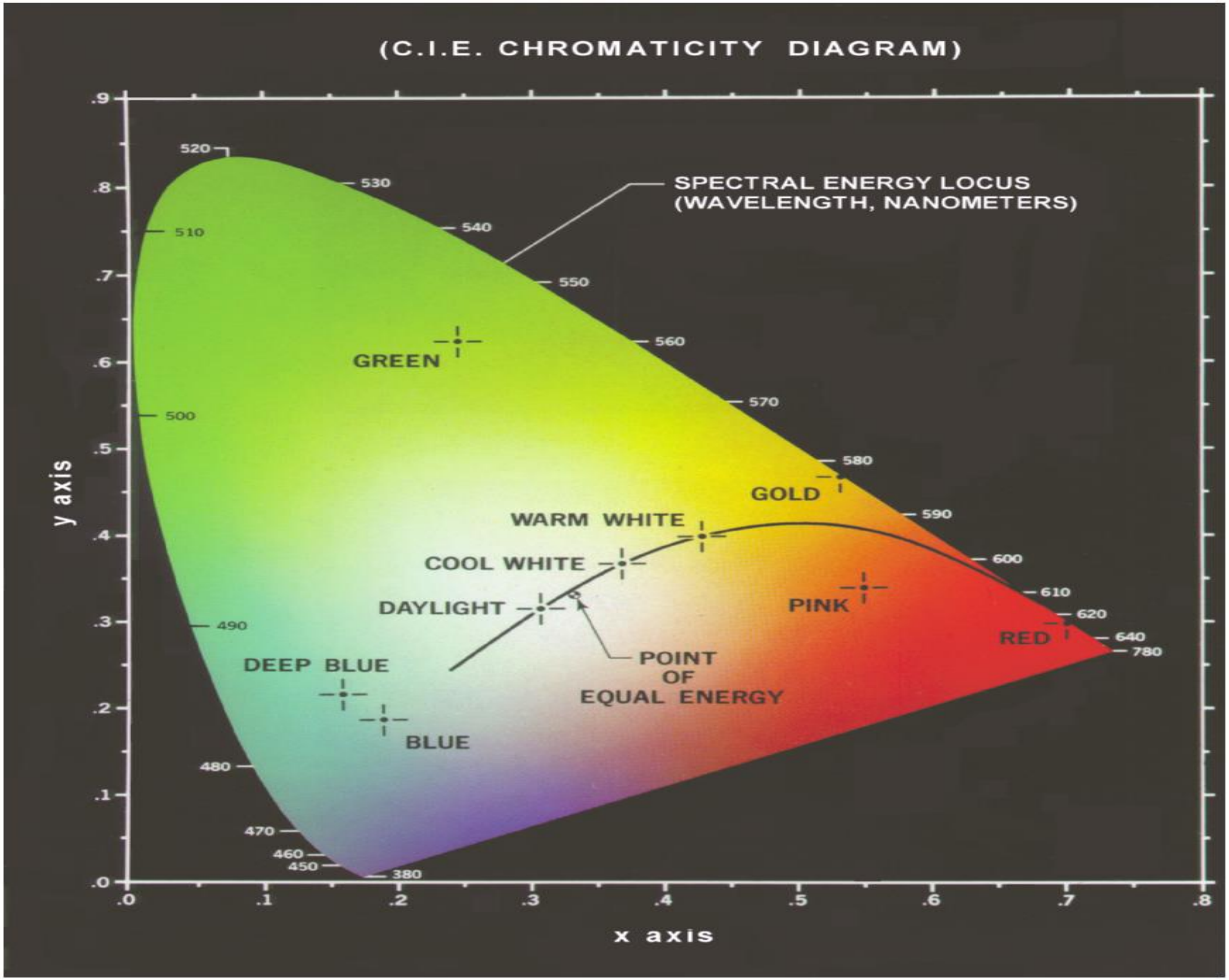
$$y = Y/(X+Y+Z)$$

$$z = Z/(X+Y+Z)$$

It is also denoted as

$$x+y+z=1 \text{(normalize form)}$$

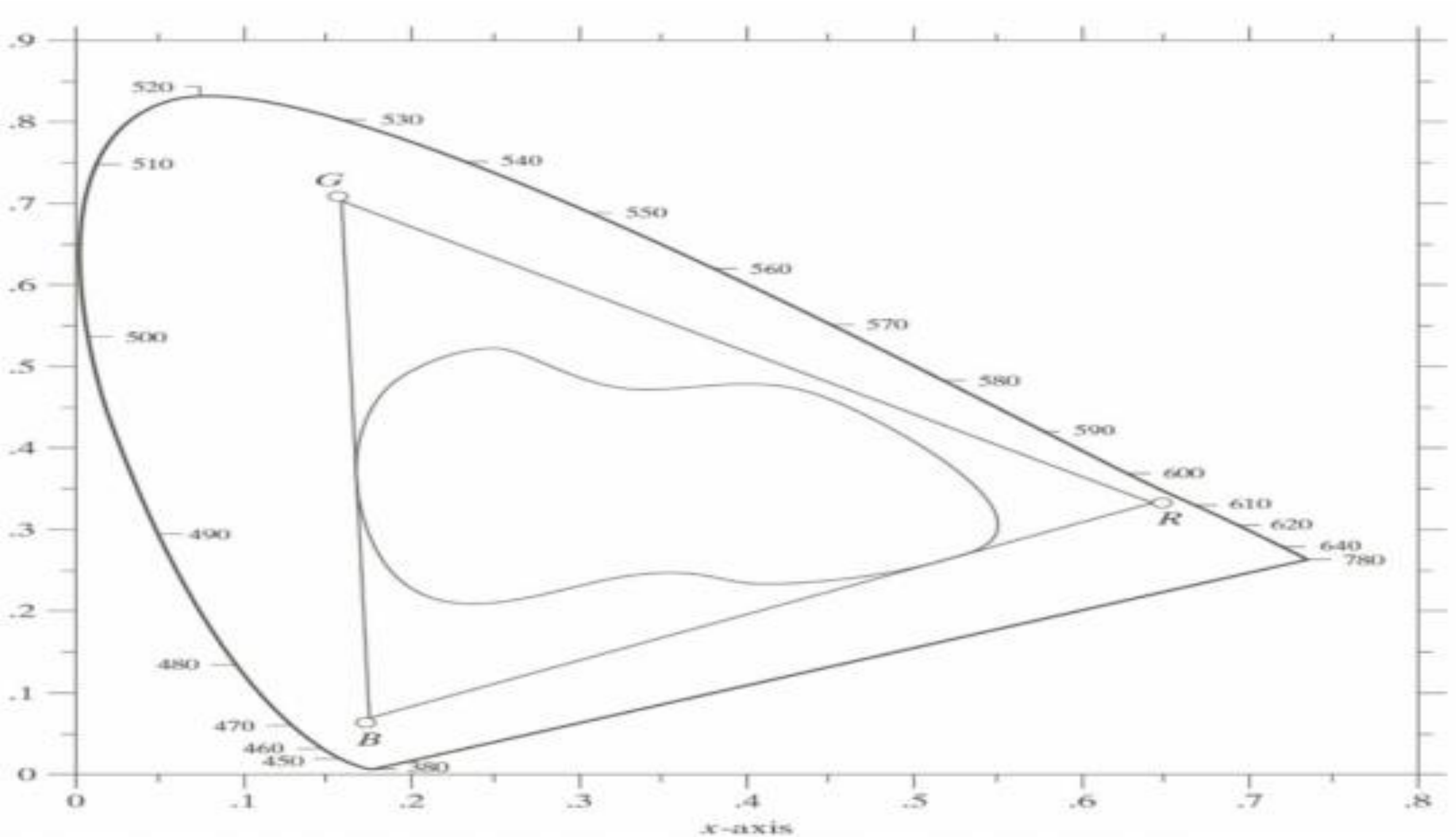
Chromaticity diagram-Colour mixing



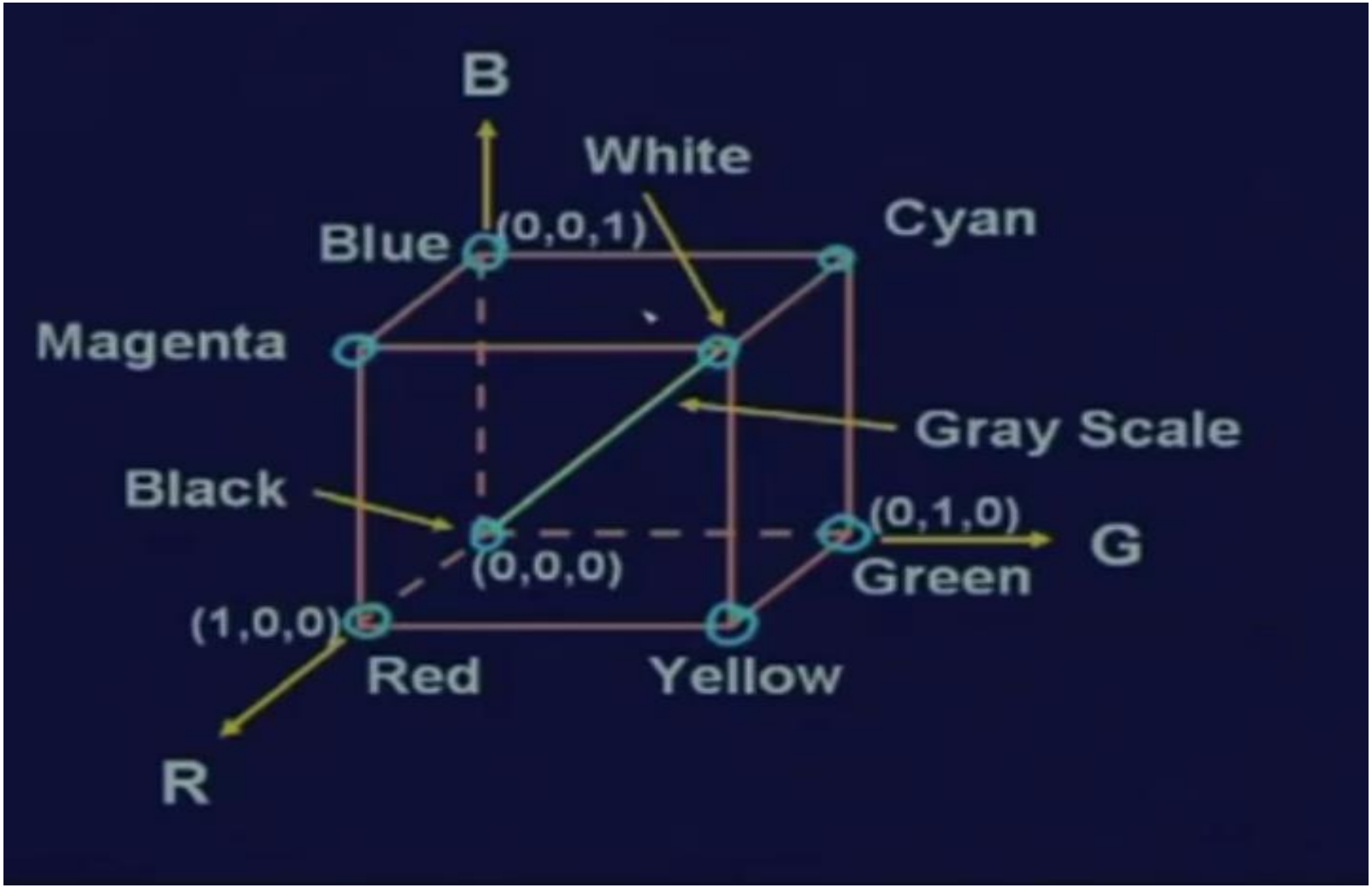
Z=1-(x+y)

C.I.E-International Commission on Illumination

CHROMATICITY DIAGRAM

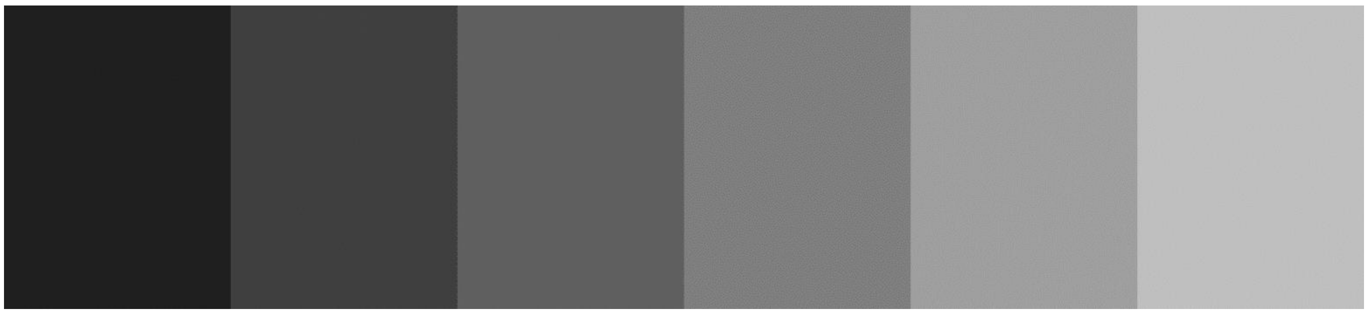


RGB COLOR CUBE-UNIT CUBE

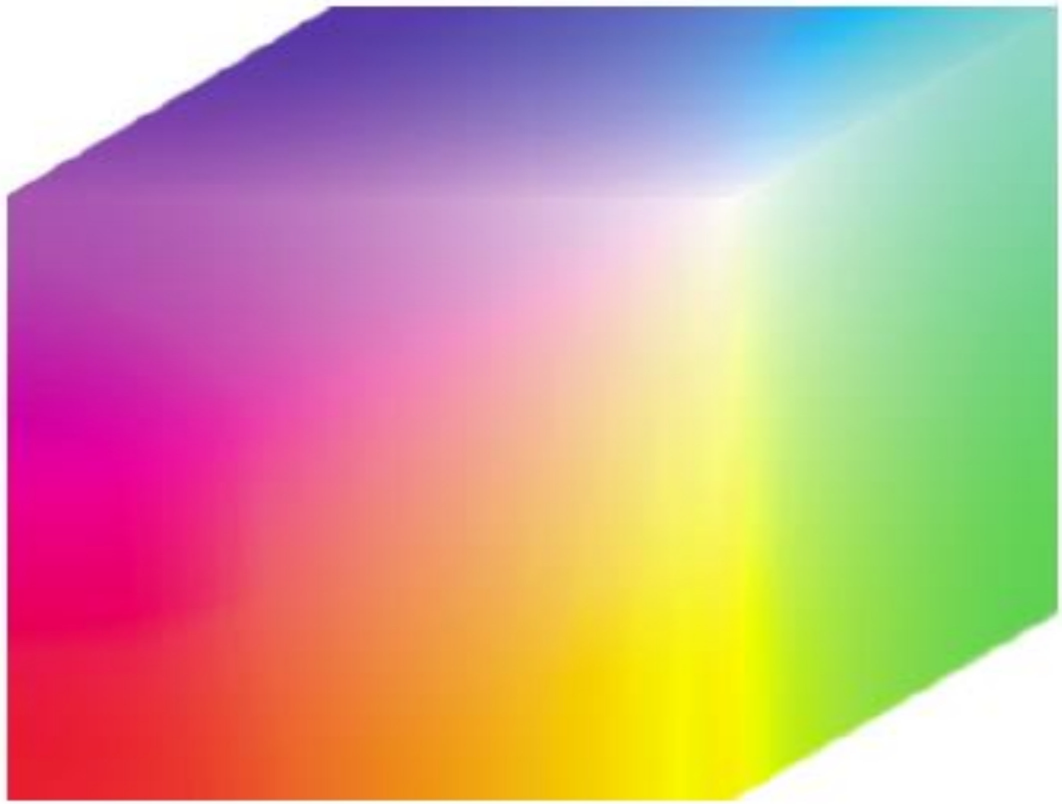


Color model is a specification of 3D- coordinates system.

Gray level band example



RGB COLOR CUBE



RGB 24-bit color cube.

RGB COLOR GENERATION

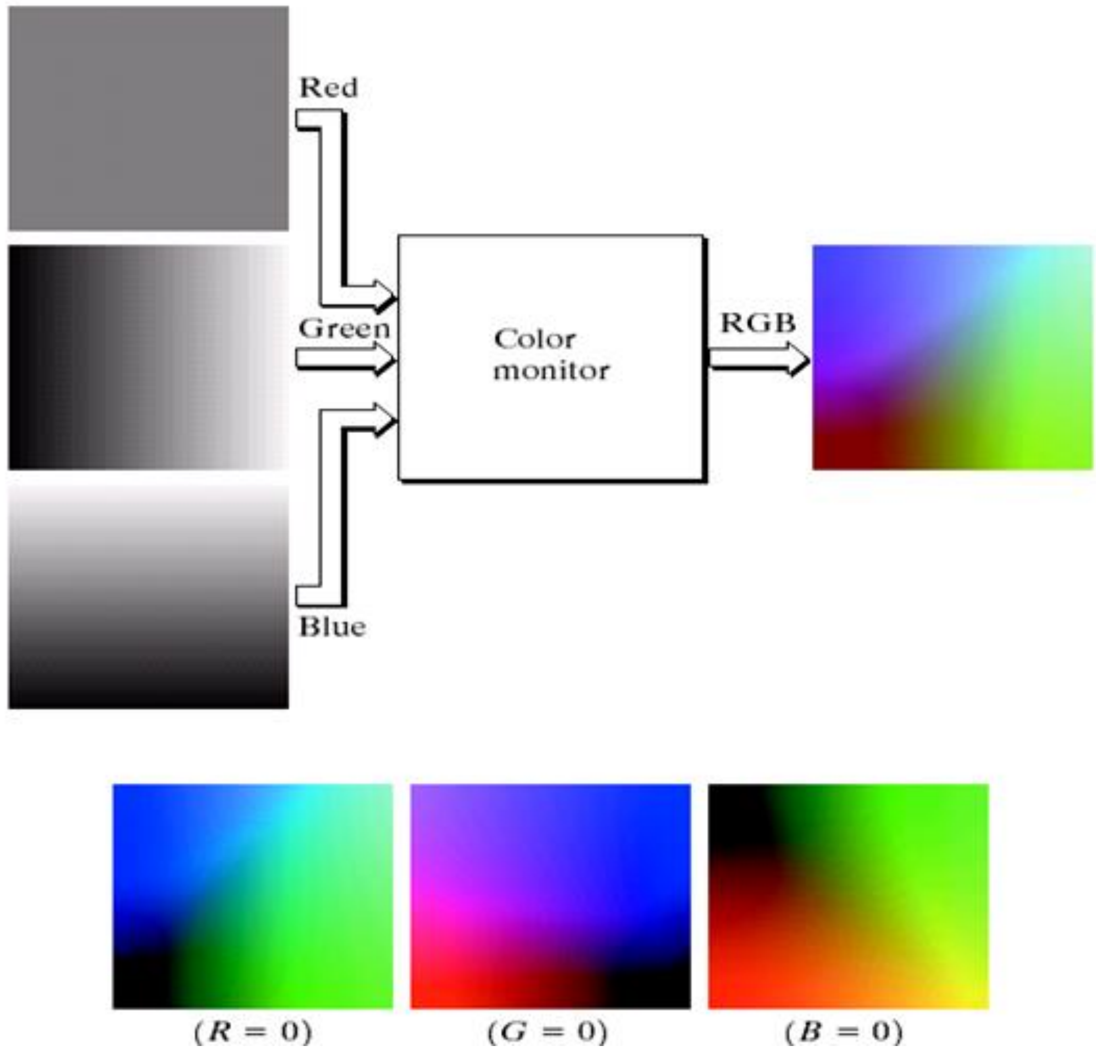
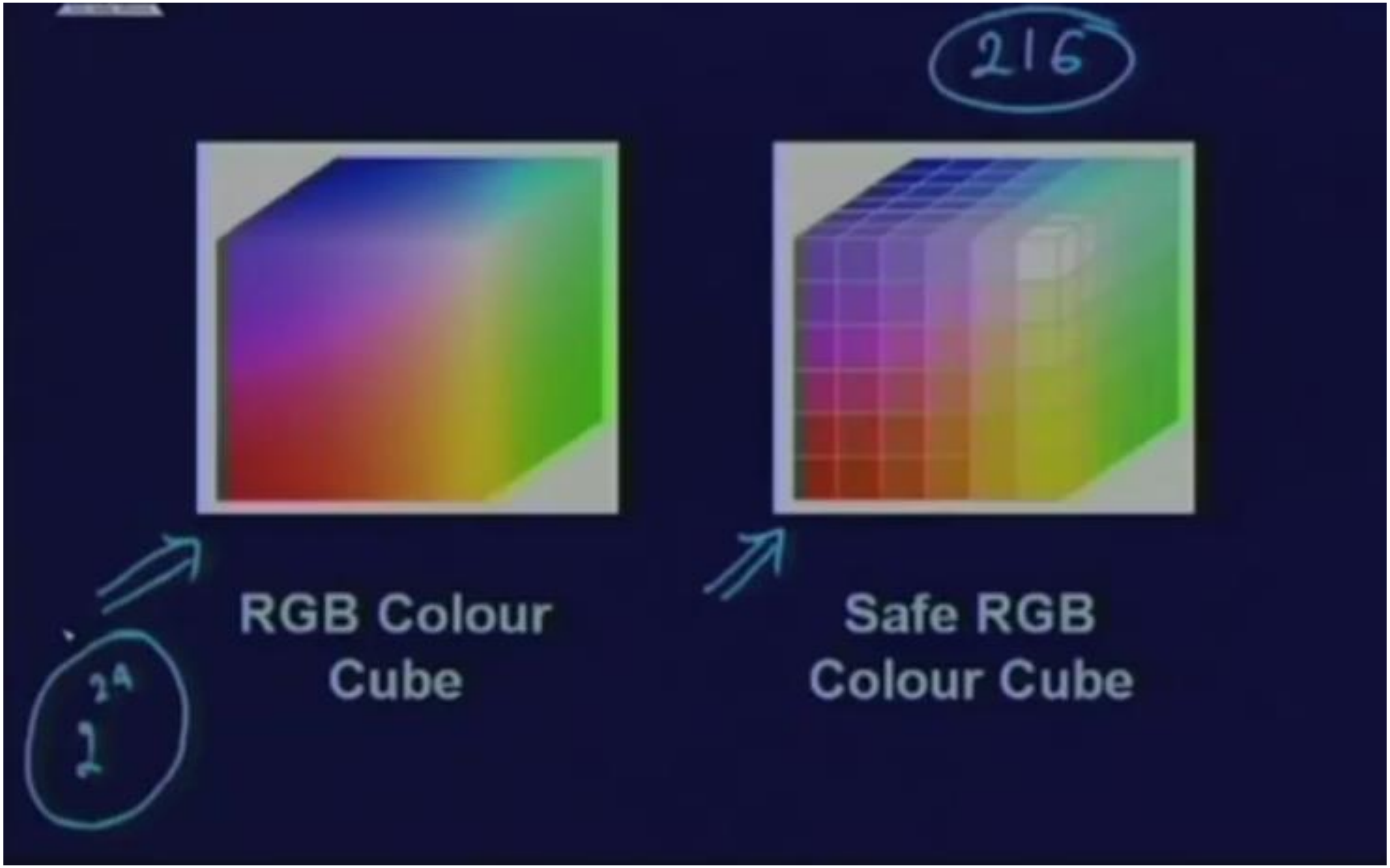


FIGURE 6.9 (a) Generating the RGB image of the cross-sectional color plane (127, *G*, *B*). (b) The three hidden surface planes in the color cube of Fig. 6.8.

RGB Color Cube and RGB safe color cube

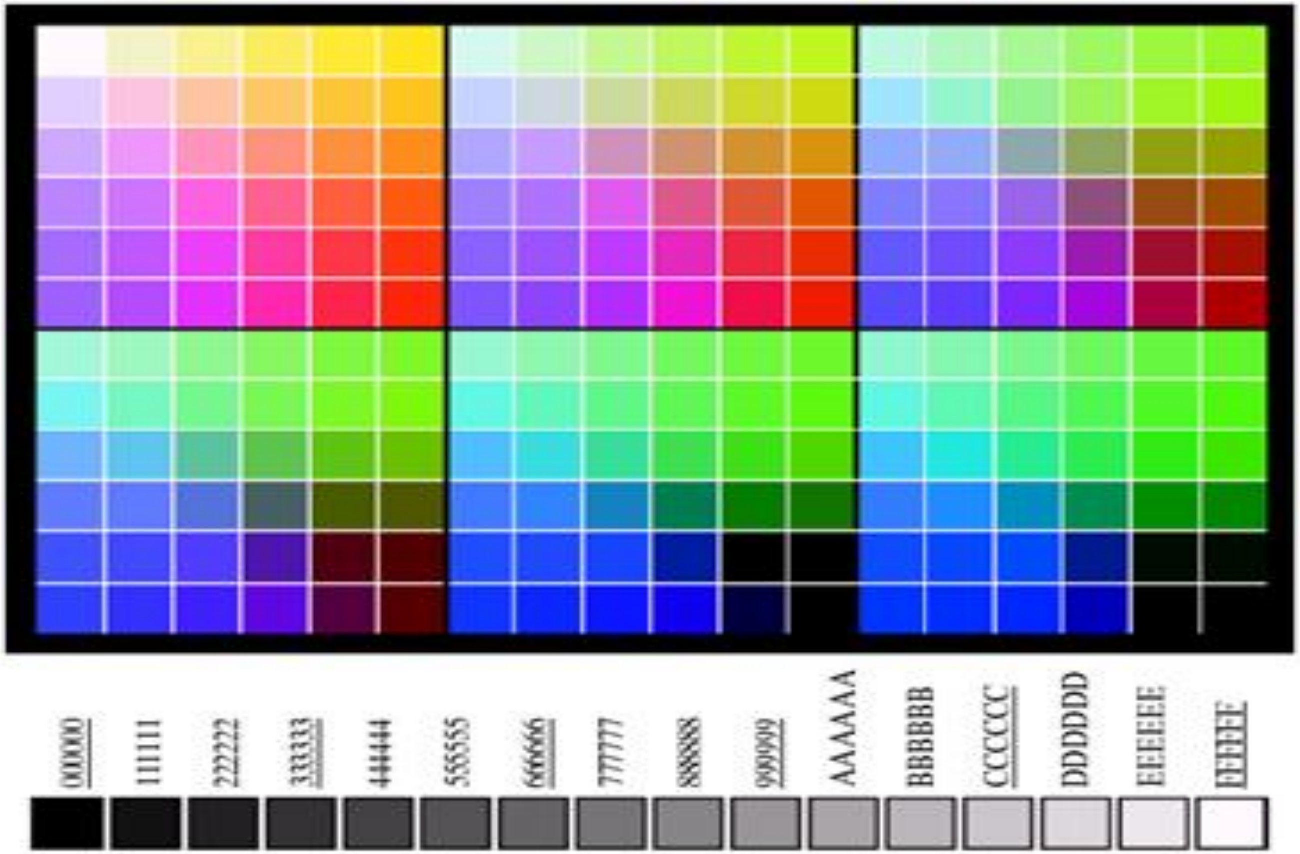


RGB safe color cube-6*6*6=216

RGB Safe Colours

Number System	Color Equivalents					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

TABLE 6.1 Valid values of each RGB component in a safe color.



a
b

FIGURE 6.10 (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

RGB

Red, Green & Blue are Primary colors.

Red, Green and Blue components of colour image

Luminance $Y=0.3R + 0.59G+ 0.11B$

The HSI Color Model

- Hue—the color
- Saturation—the amount of white that is mixed with the hue
- Intensity—expresses the brightness or luminance of the chromaticity (=hue and saturation)

CONVERTING FROM RGB TO HSI

$$H = \begin{cases} \theta, & \text{if } B \le G \\ 360 - \theta, & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2 + (R-B)(G-B)}} \right)$$

$$S = 1 - \frac{3\min(R,G,B)}{(R+G+B)}$$

$$I = \frac{(R+G+B)}{3}$$

Prof(Dr) P P Ghadekar

CONVERTING FROM HSI TO RGB CASE R-G

$$\text{Given } 0 \le H, S, I \le 1, \text{ rescale } H = 360° H$$

$$\textbf{Case R-G}, (0° \le H < 120°):$$

$$B = I(1-S)$$

$$R = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

$$G = 3I - (R+B)$$

$$\text{Note: } 0 \le R, G, B \le 1 \text{ may be rescaled as needed.}$$

CONVERTING FROM HSI TO RGB CASE G-B

Given $0 \le H, S, I \le 1$, rescale $H = 360° H$

CaseG-B, $(120° \le H < 240°)$: Compute $H = H - 120°$

$$R = I(1 - S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

$$B = 3I - (R + G)$$

Note: $0 \le R, G, B \le 1$ may be rescaled as needed.

Prof(Dr) P P Ghadekar

CONVERTING FROM HSI TO RGB CASE B-R

Given $0 \le H, S, I \le 1$, rescale $H = 360° H$

Case B-R, $(240° \le H \le 360°)$: Compute $H = H\text{-}240°$

$$G = I(1 - S)$$

$$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

$$R = 3I - (B + G)$$

Note: $0 \le R, G, B \le 1$ may be rescaled as needed.

Prof(Dr) P P Ghadekar

YCbCr

The Human visual system is less sensitive to colour than to Luminance.

Luminance $Y=K_rR + K_gG + K_bB$

Where K are weighting factors

The color information can be represented as colour difference components

$$Cb=B - Y$$

$$Cr =R - Y$$

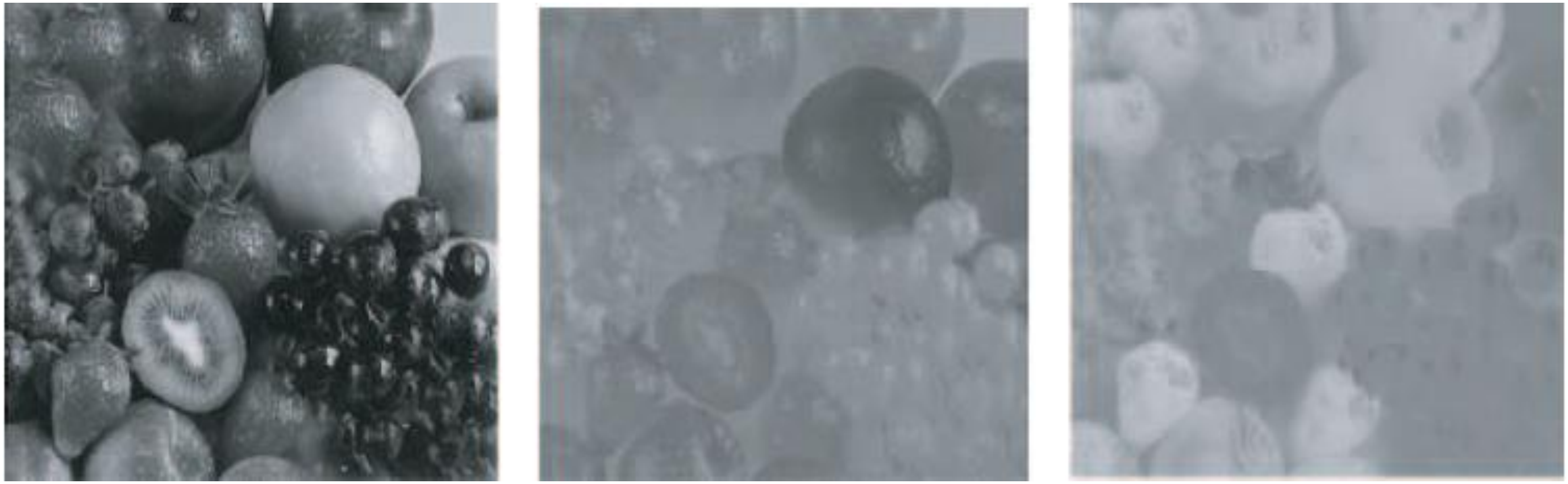
$$Cg=G - Y$$

YCbCr

Cr, Cg and Cb components

However Cb + Cr + Cg is constant. So only two components need to be stored or transmitted

YCbCr



(A) (B) (C)

(A) Luminance (Y), (B) blue-yellow (Cb), (C) red-green (Cr) components.

YCbCr Sampling Formats

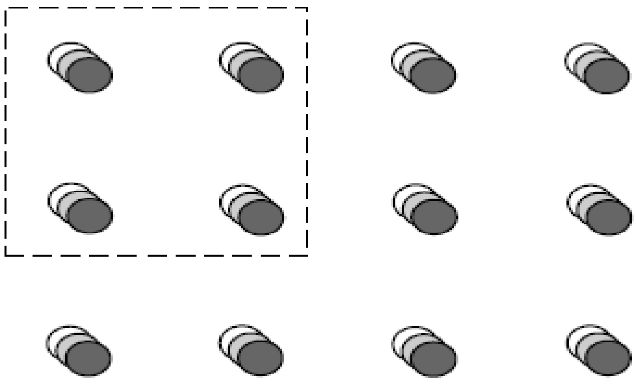
There are three sampling formats

4:4:4 - Y, Cb & Cr have same resolution.

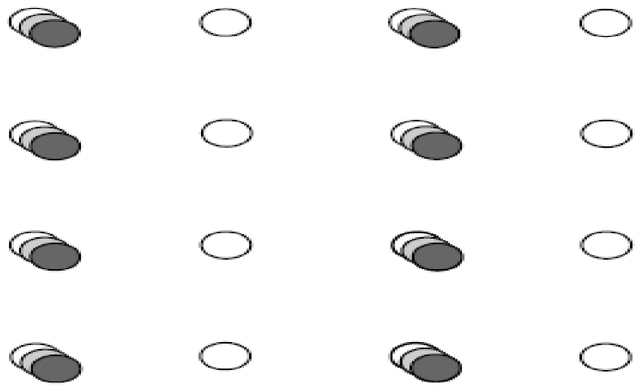
4:2:2 - Cb, Cr have same vertical resolution as the luma but half the horizontal resolution.

4:2:0 - Cb, Cr each have half the horizontal and vertical resolution of Y.

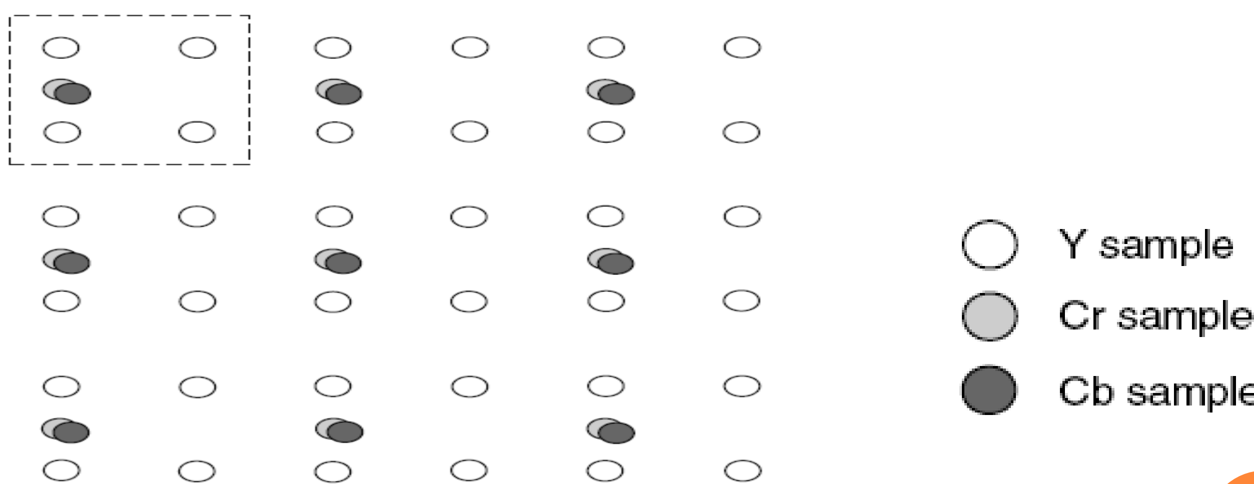
YCbCr Sampling Patterns



4:4:4 sampling -12 samples



4:2:2 sampling – 8 samples



4:2:0 sampling – 6 samples

Images obtained by modifying luminance and chrominance channels.

(a)

(b)

(c)

(d)

(a) The luminance channel is blurred and the chrominance channel is left unmodified; (b) The chrominance channel is blurred and the luminance channel is left unmodified; (c) The luminance channel is down sample and the chrominance channel is left unmodified; (d) the chrominance channel is down sample and the luminance channel is left unmodified.

CMY MODEL

- Relationship between colors from the CMY and RGB models:

  C=1-R M=1-G Y=1-B

- Problem in the CMY model is inability to reproduce large number of colors.
- It is difficult to reproduce fluorescent and similar colors and the most importantly the black!!!
- Approximately just about 1 million of colors is possible to reproduce with good quality.
- The main problem is black since it is very important for human (important for image recognition, edges of shapes, etc).
- Reason: mix cyan, magenta and yellow and you will get dark but not black color (or you will use huge amount of color what is not economical).

CMY MODEL

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Black Colour-Muddy Black colour.

CMYK- Colour model

CMYK MODEL

- Printing machines usually print images in the CMYK model (4-channel model). K means blacK since B is reserved for Blue.
- Relationship between CMY and CMYK models:

$K=\min(C,M,Y)$

$C'=C-K$

$M'=M-K$

$Y'=Y-K$

CMY & CMYK MODEL – COMPARISON

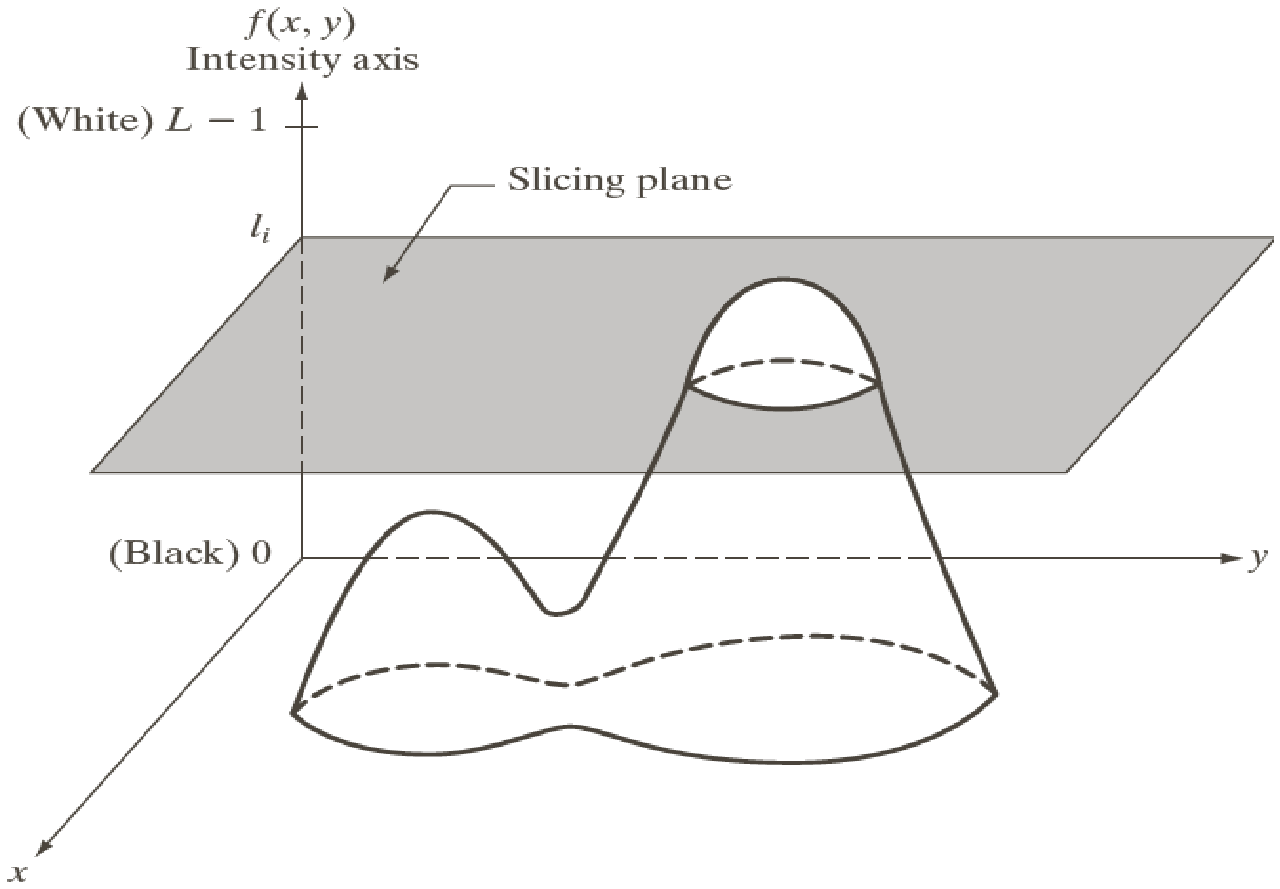
CMYK image

CMY image

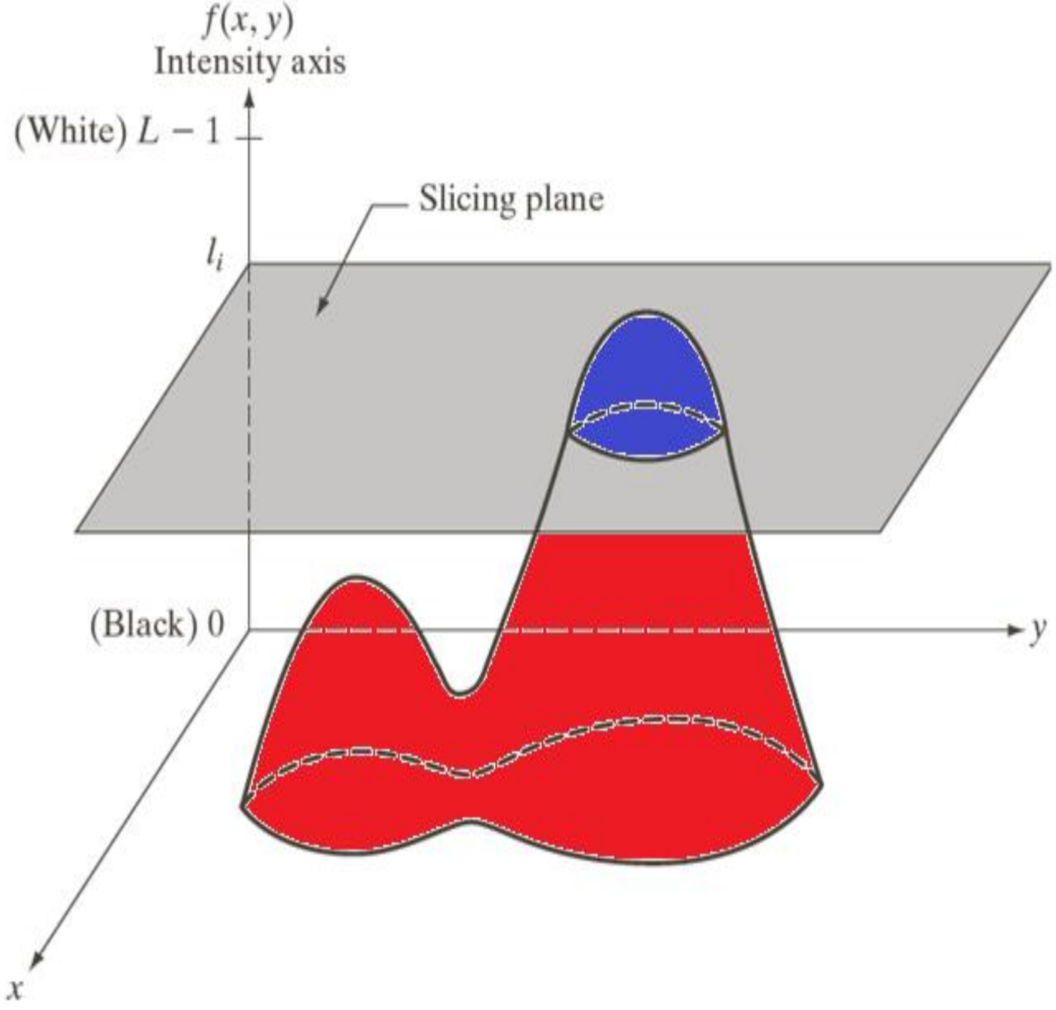
Colour IMAGE PROCESSING

- Full colour image processing
- Pseudo colour processing or False colour processing

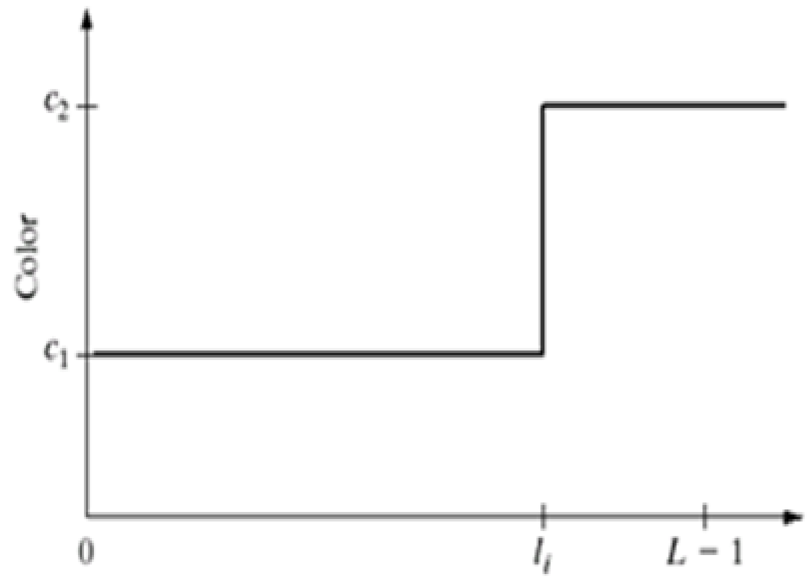
PSEUDO COLOURING
INTENSITY SLICING TECHNIQUE



Pseudo colouring
Intensity slicing technique



INTENSITY SLICING TECHNIQUE



Gray levels

- An Alternative representation of Intensity slicing Technique

Pseudo colouring

0 ------ $L-1 \longrightarrow$ L no of intensity values.

$L_0 \longrightarrow$ black [$f(x , y) = 0$]

$L_{L-1} \longrightarrow$ White [$f(x , y) = L-1$]

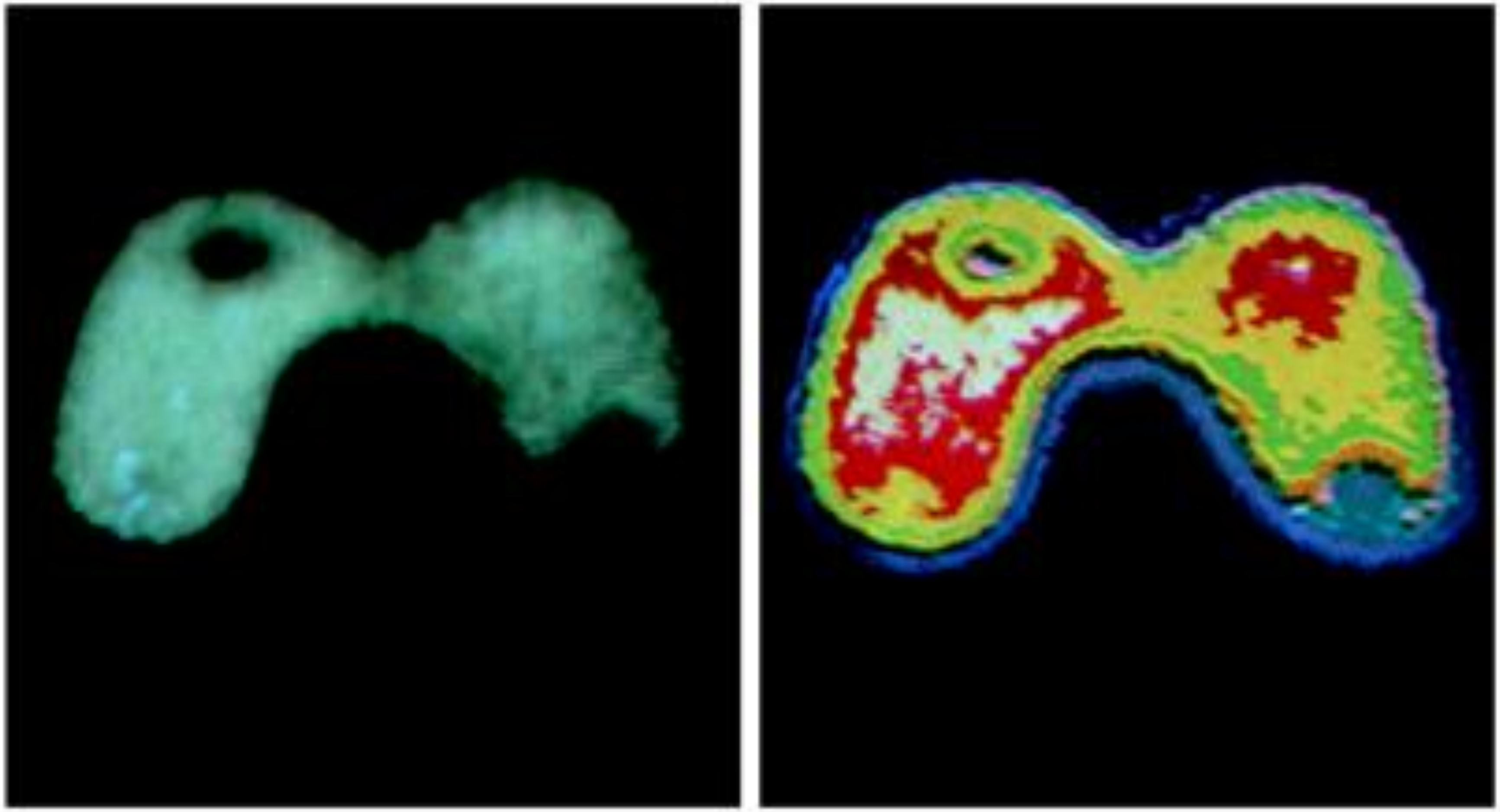
P no of planes perpendicular to the intensity axis

$l_1, l_2,$------ l_p

$0 < P < L-1$

$P+1 \longrightarrow$ no of intervals.

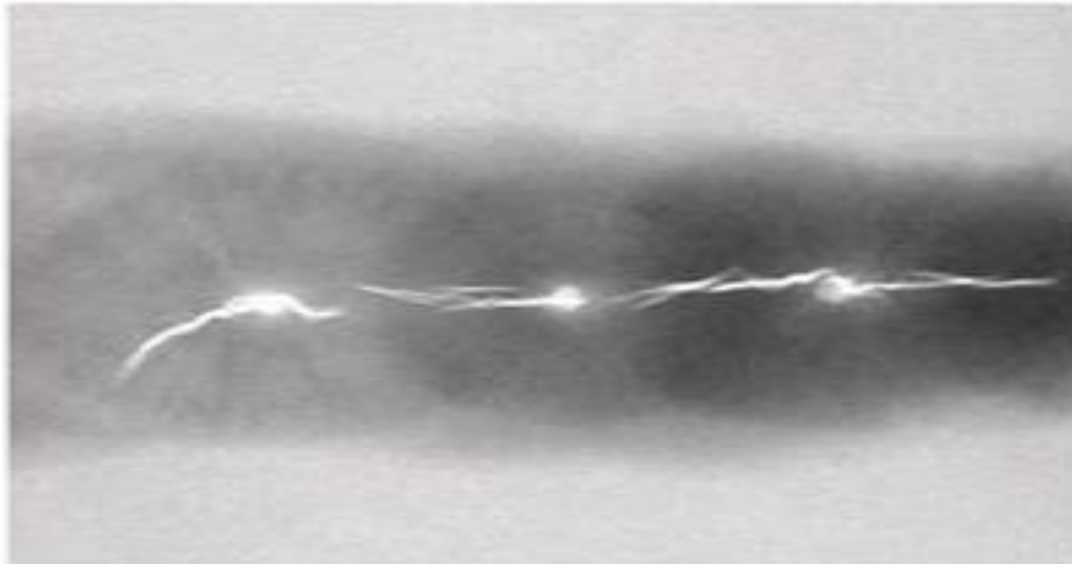
Pseudo colouring



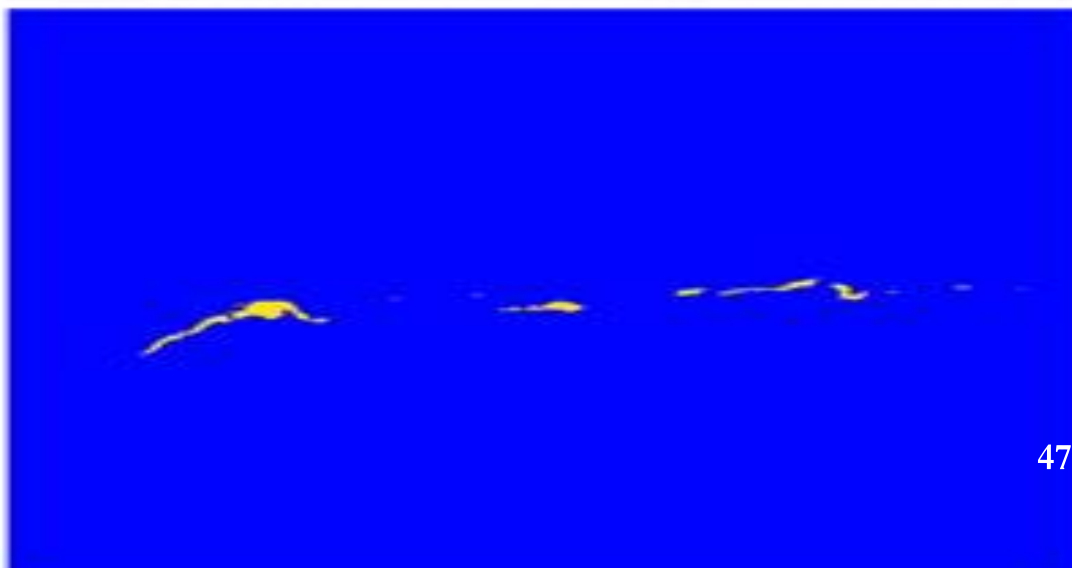
(a)Monochrome image of the Picker Thyroid Phantom (b) Result of the Intensity Slicing into eight colours.

PSEUDO COLOURING

(a) Monochrome X-ray Image of a weld



(b) Result of colour coding



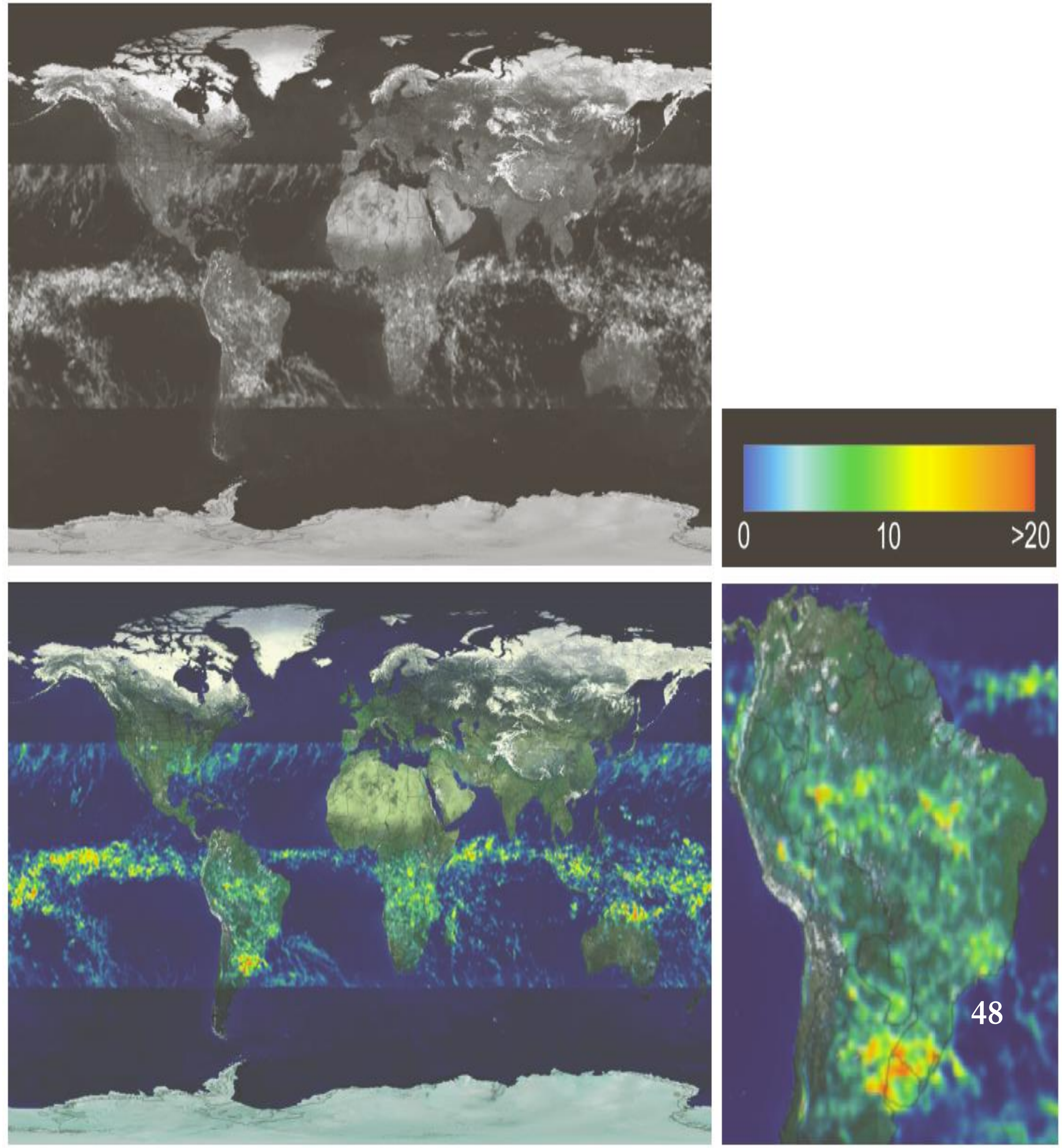
Pseudo colouring

(a) Gray scale image in which intensity corresponds to the average monthly rainfall.

(b) Colours assigned to intensity values

(c) Colour coded image

(d) Zoom of the South America region

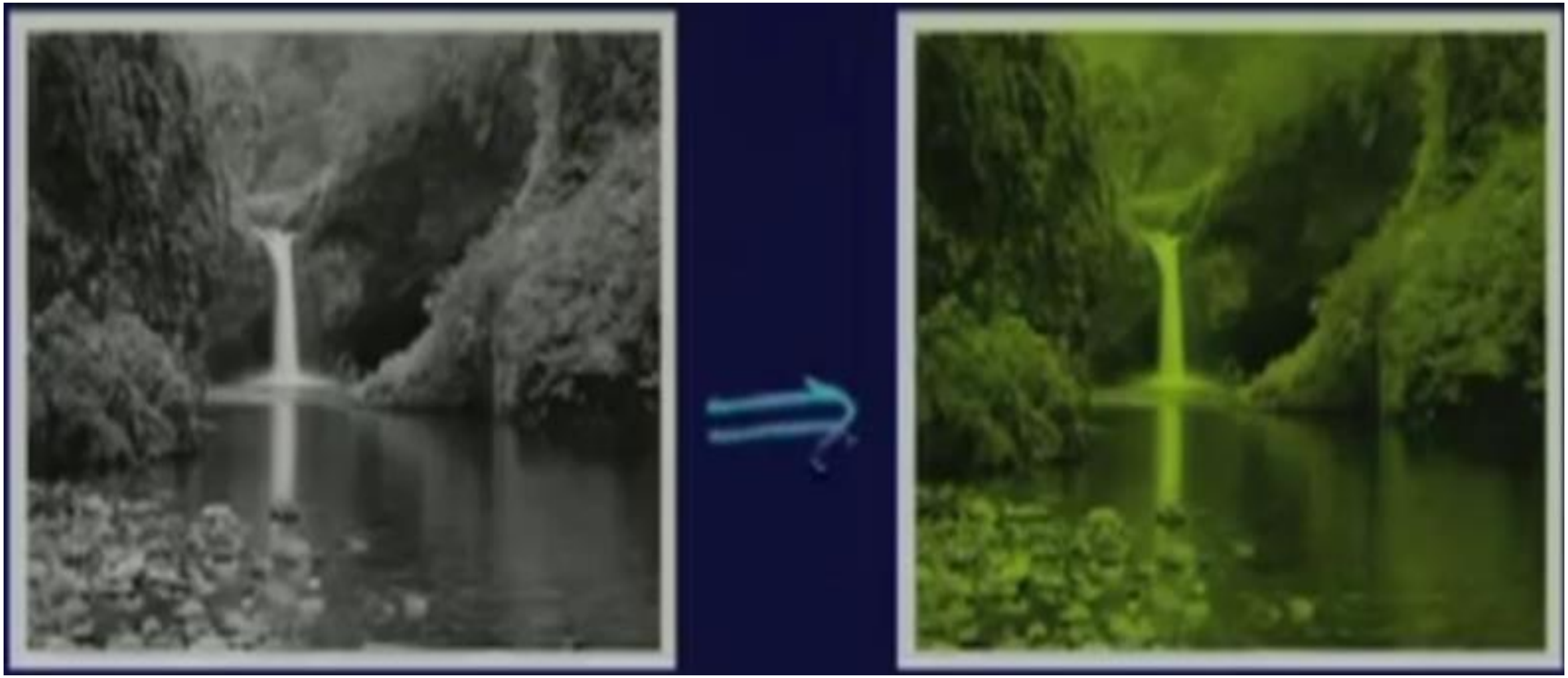


GRAY TO COLOUR CONVERSION

Gray to colour conversion

$$f_R(x,y) = f(x,y)$$
$$f_G(x,y) = 0.33\ f(x,y)$$
$$f_B(x,y) = 0.11\ f(x,y)$$

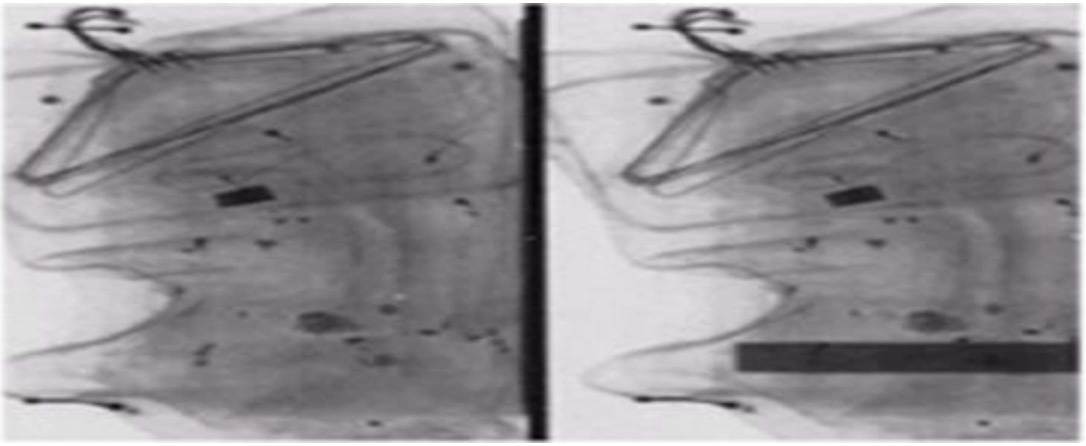
Gray to colour conversion

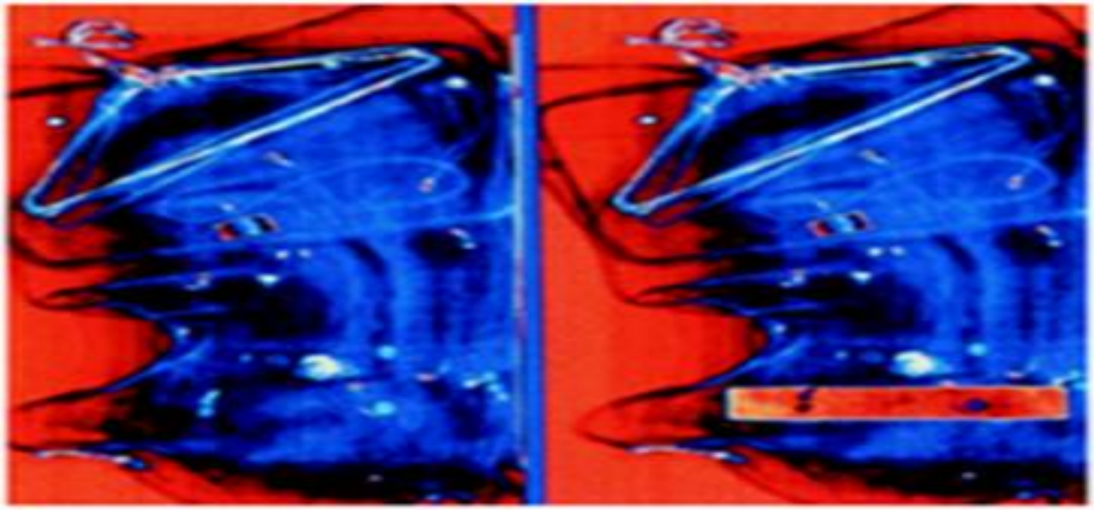


$$f_R(x,y) = 0.33\ f(x,y)$$
$$f_G(x,y) = f(x,y)$$
$$f_B(x,y) = 0.11\ f(x,y)$$

Pseudo colour enhancement

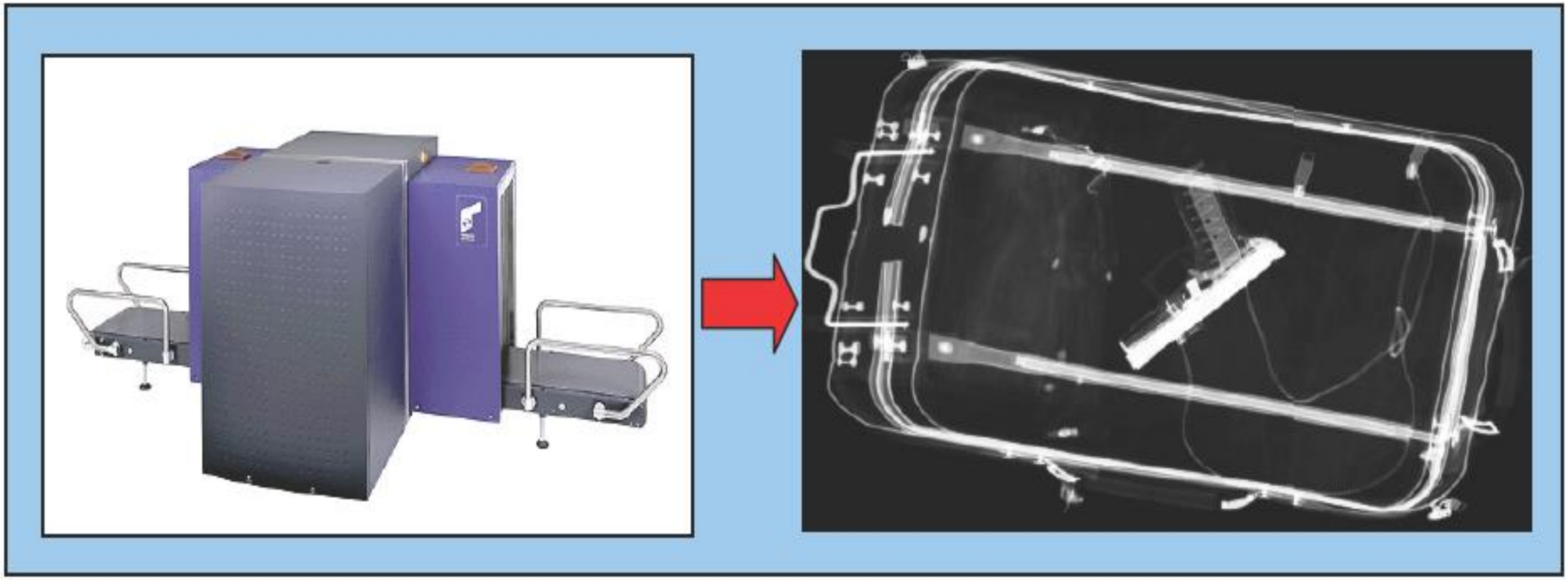
(a) Two monochrome images of luggage obtained from an airport x-ray scanning system





(b) Images obtained with transformation function

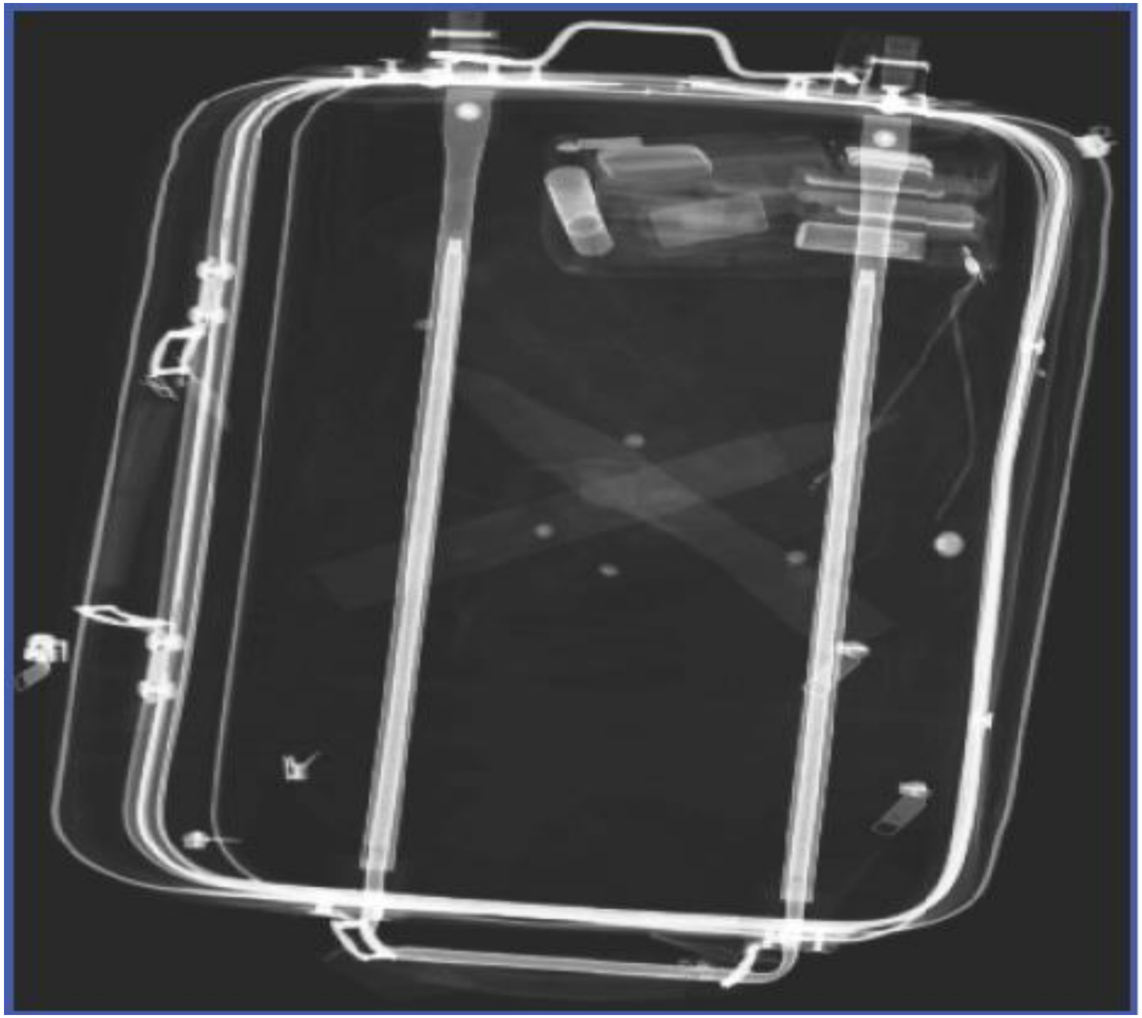
Pseudo colour enhancement



Inspection Module **Sample output image**

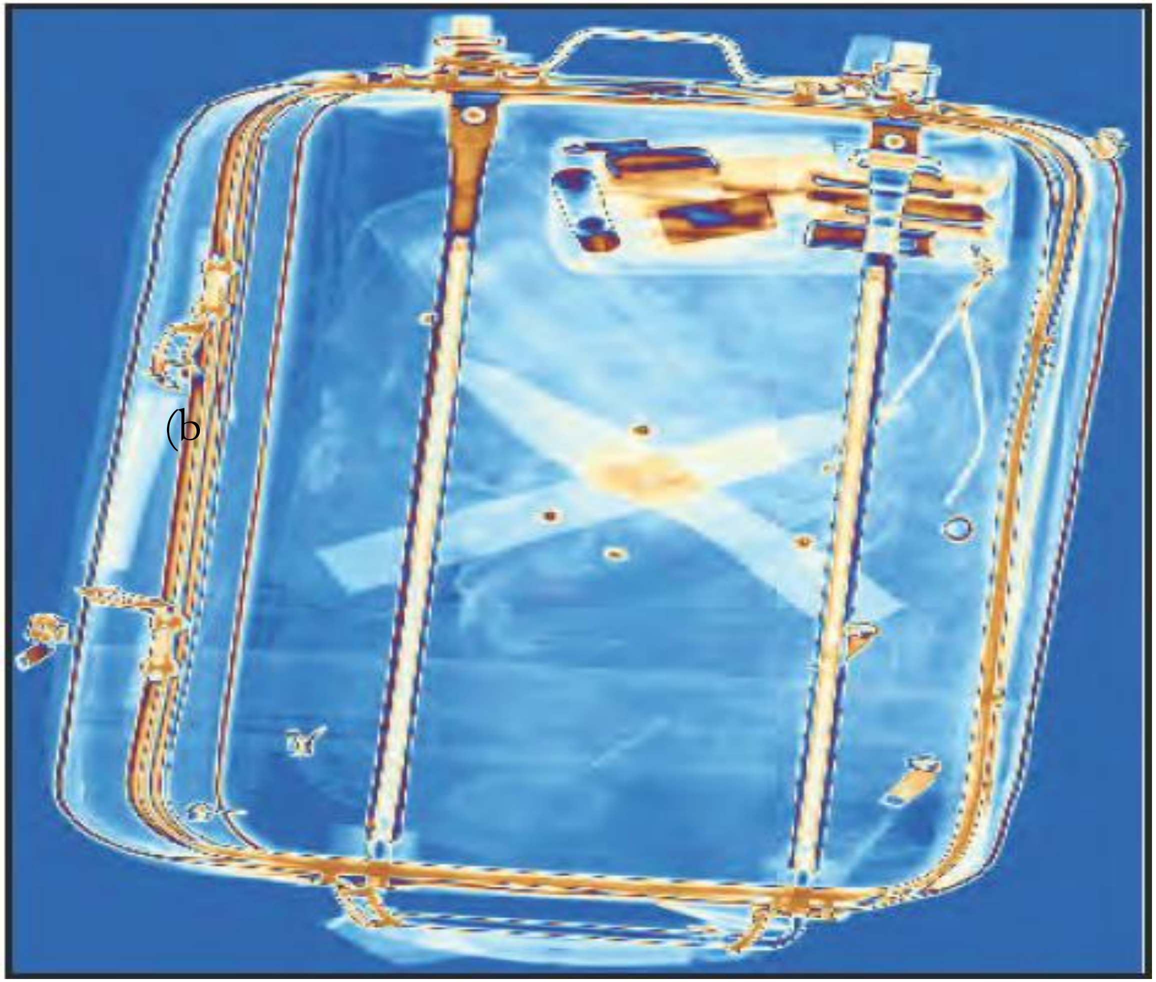
Luggage inspection module and a sample x-ray scan containing a threat object

Pseudo colour enhancement

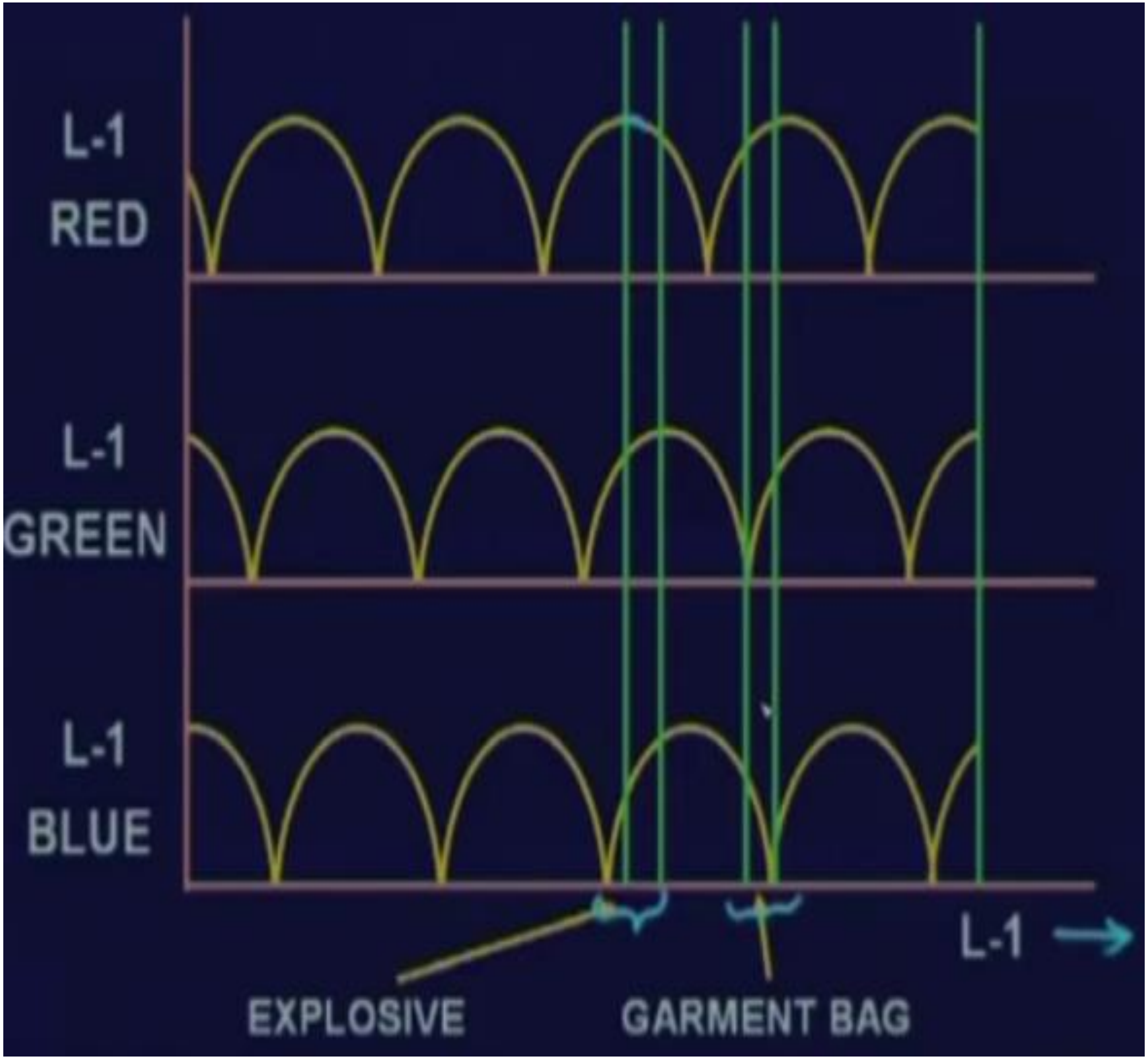


(a) Original grayscale image

Pseudo colour enhancement

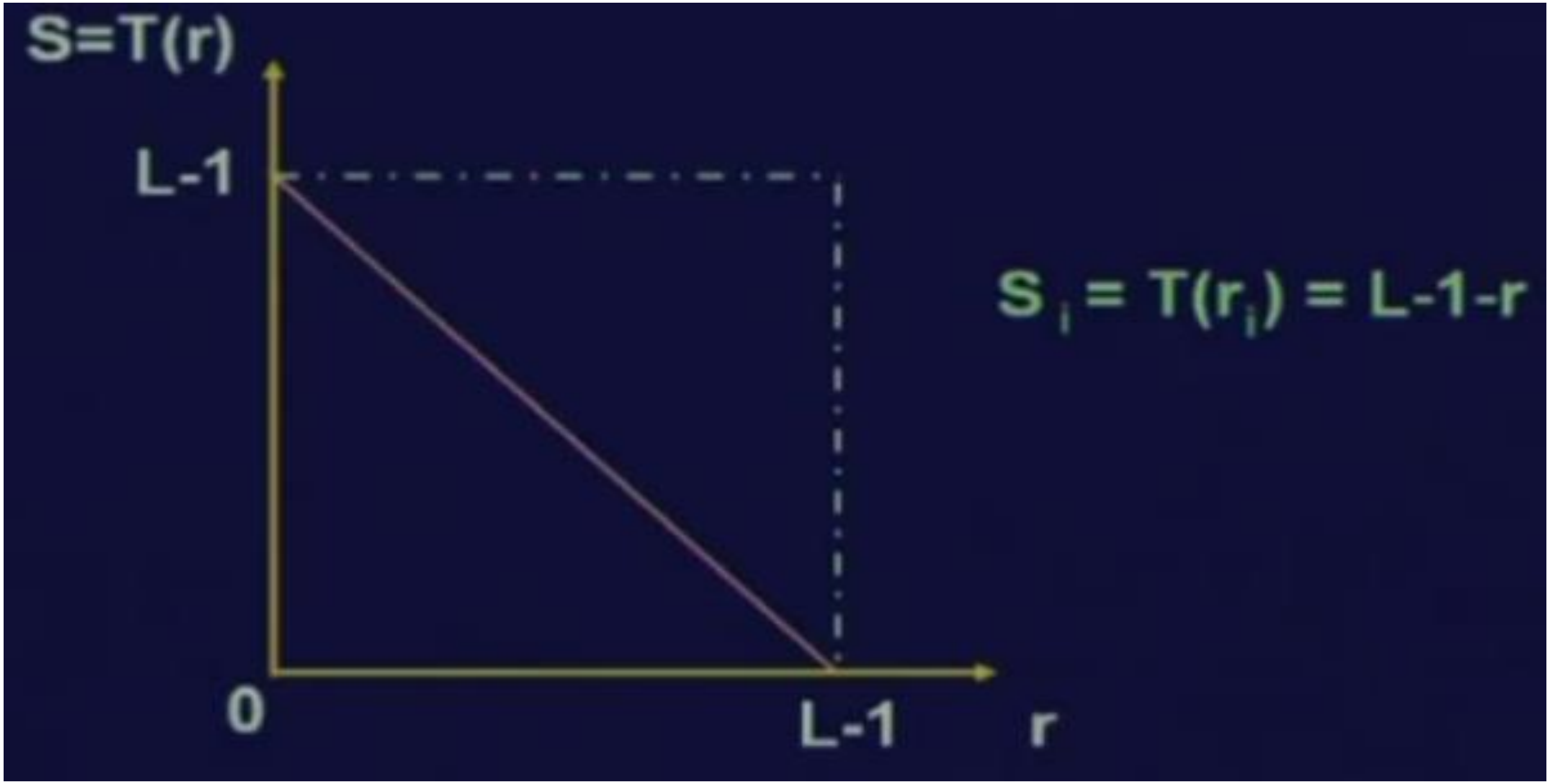


(b) Output image obtained after applying sine transformation clearly showing the threats (dark red and purple glass knives)



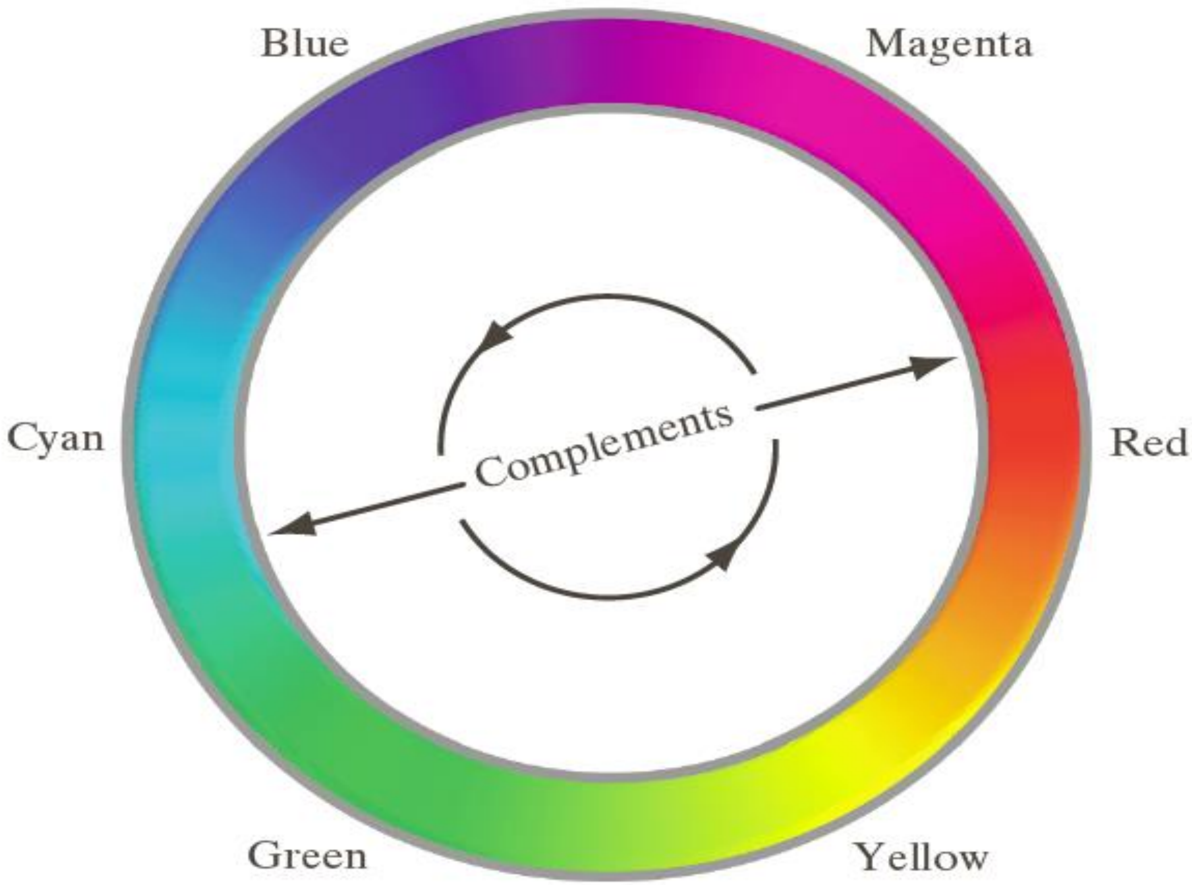
Transformation Functions used to obtain the images shown in previous slide

Colour Compliment



Where L-number of intensity values, r intensity of original image

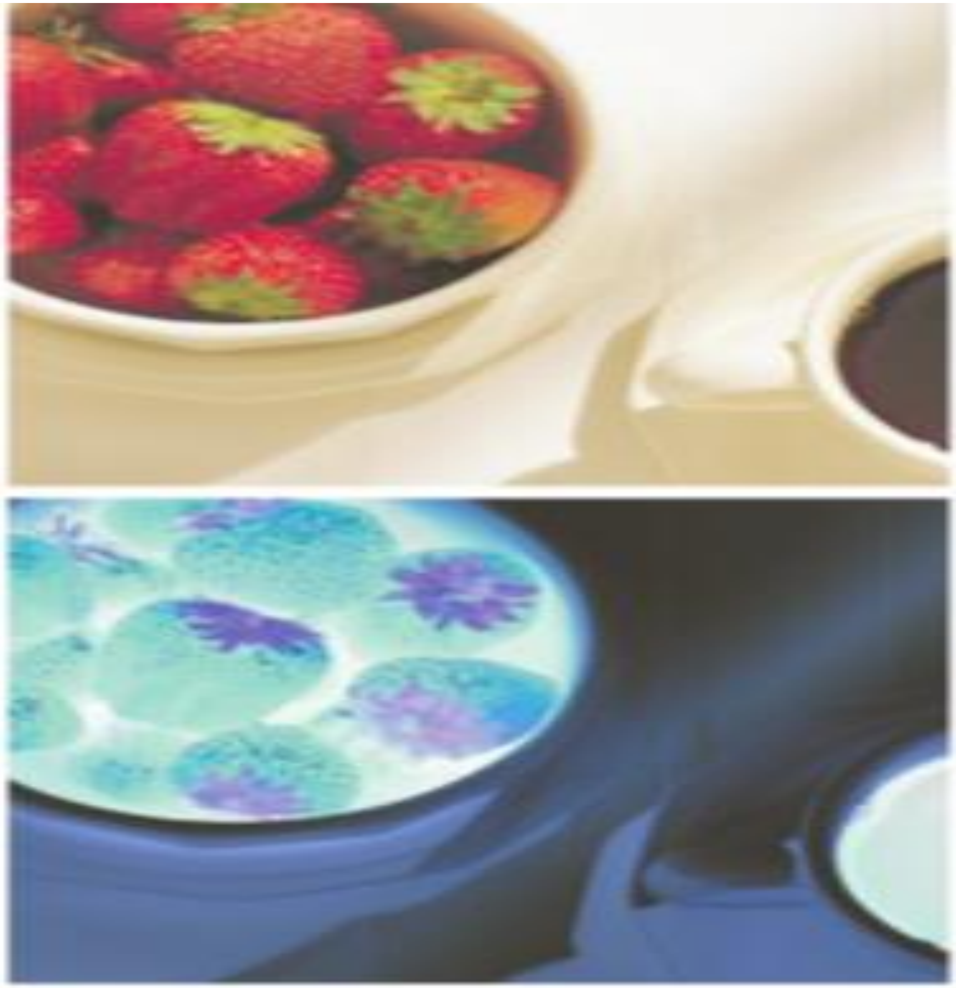
Colour Compliment



- It is analogous to gray-scale image negation operation.

Colour Compliment-EXAMPLE

(A)



(B)

(C)

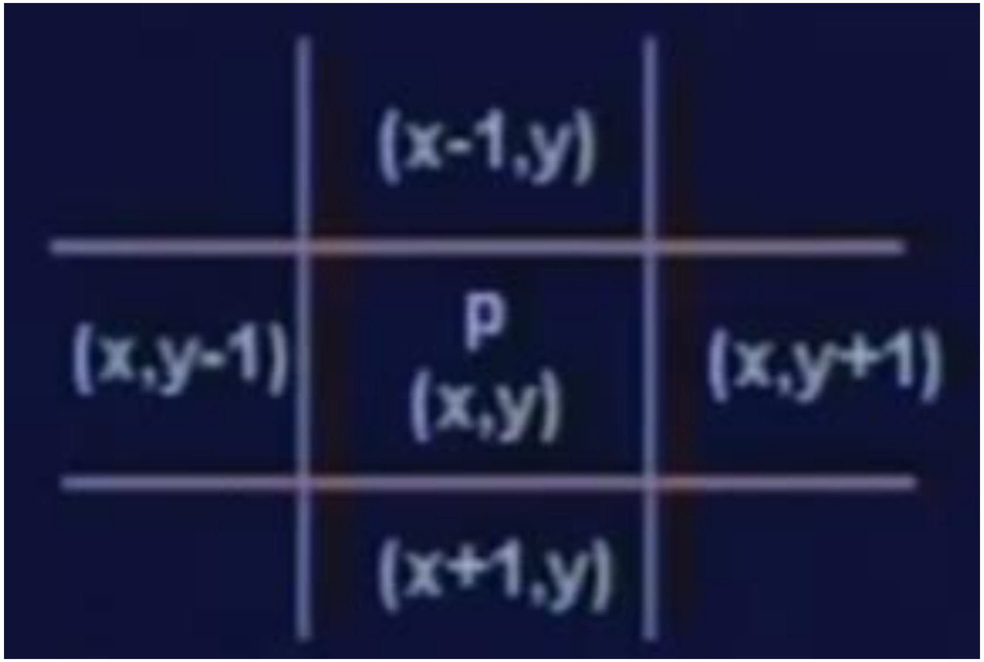
(A)-Original Image, (B) Colour compliment of given Image, (C) Transformation Function for RGB.

RELATIONSHIP BETWEEN PIXELS

- On the completion the student will be able to
- Explain what is pixel neighborhood and different types of neighborhood.
- Explain what is meant by connectivity.
- Explain what is adjacency and different types of adjacency.
- Learn different distance measure.

NEIGHBORHOOD OF A PIXEL

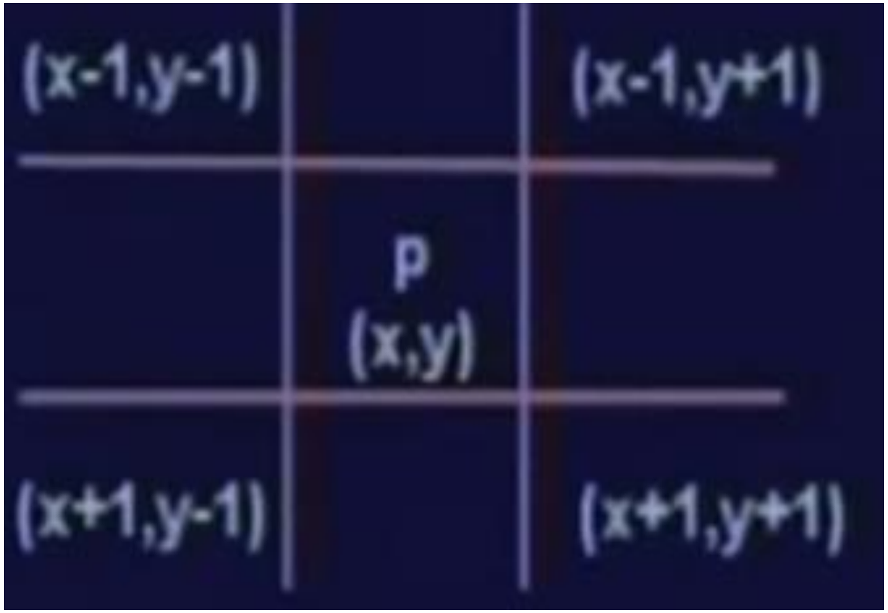
A pixel p at location (x,y) has two horizontal and two vertical neighbours.



This set of four pixels is called 4-neighbours of p=N_4(p).
Each of these neighbours is at a unit distance from p.
If p is a boundary pixel then it will have less Number of neighbours.

DIAGONAL AND 8-NEIGHBORS

A pixel p has four diagonal neighbours= $N_D(p)$

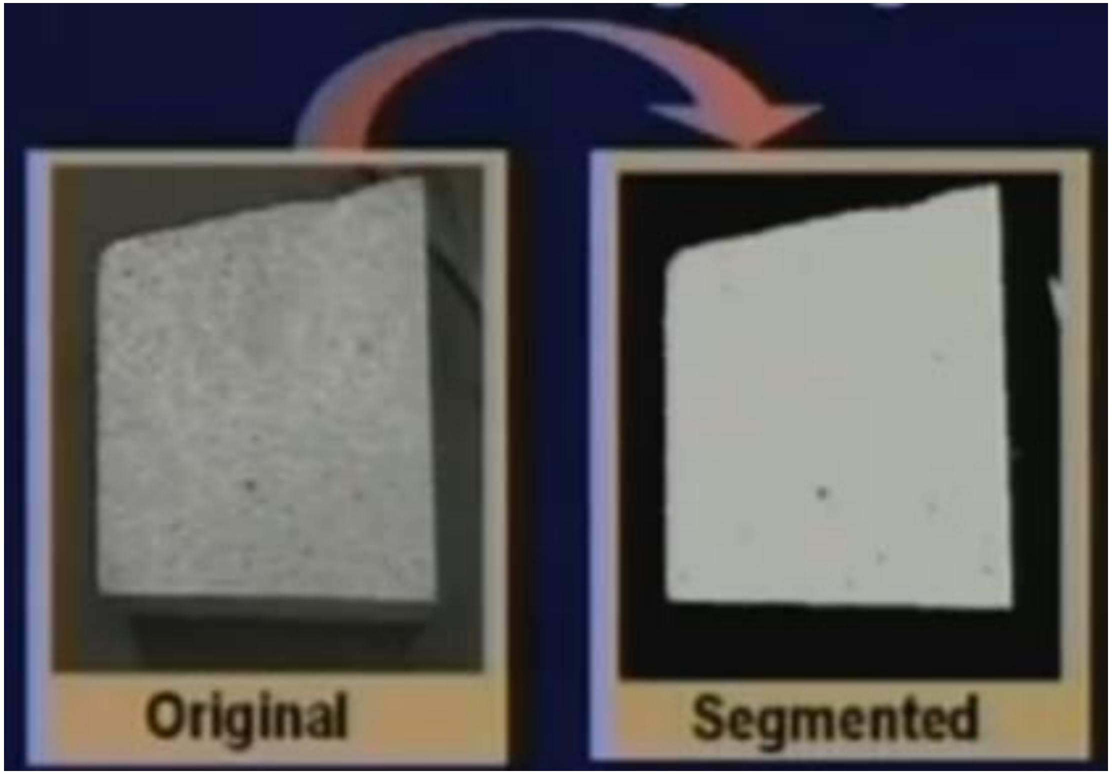


The Point of $N_4(p)$ and $N_D(p)$ together are called 8-neighbours of p.
$N8(p)= N_4(p) \cup N_D(p)$.
If p is a boundary pixel then both $N_4(p)$ and $N_D(p)$ will have less number of pixels.

Pixel connectivity

Connectivity between pixels is a very Important concept.
It is very useful for

- Establishing object boundaries.
- Defining Image components/regions etc.



If f(x,y) > Threshold
(x,y) Є Object
Else
(x,y) Є Background

What is connectivity

Two pixels are said to be connected if they are adjacent in some sense

- They are neighbours (N_4, N_D or N_8)and
- Their intensity values (gray levels) are similar.

For a Binary image B, two points p and q will be connected if q $\in N(p)$ or p $\in N(q)$ And $B(p) = B(q)$.

CONNECTIVITY FOR GRAY SCALE IMAGE

Let V be the set of gray levels used to define Connectivity for two points **$f(p),f(q) \in V$,** three types of Connectivity are defined.

- **4-Connectivity => $p , q \in V$ & $p \in N_4(q)$**
- **8-Connectivity => $p , q \in V$ & $p \in N_8(q)$**
- **M-Connectivity => (Mixed Connectivity)**

$p , q \in V$ are M-connected if

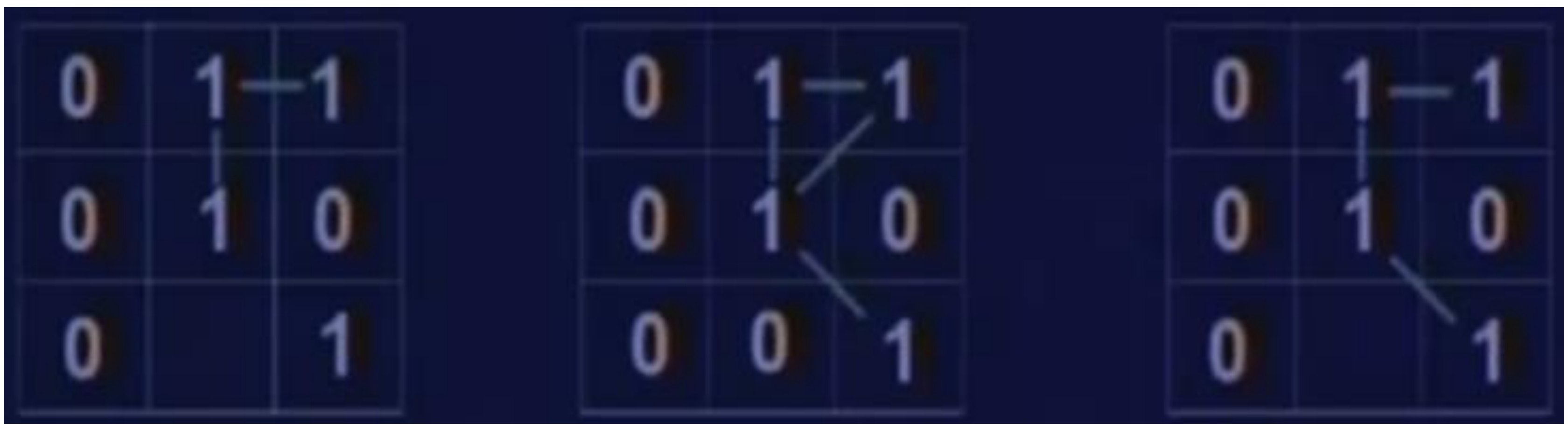
i. $q \in N_4(p)$ Or

ii. $q \in N_D(p)$ and the $N_4(p) \cap N_4(q)$ => empty,

Set of pixels that are 4-neighbours of both p and q and whose values are from V

CONNECTIVITY

Mixed connectivity is a modification of 8-connectivity.
Eliminates multiple path connections that often arise with 8-connectivity. Pixels should not have common points.
Example-V{1}



4-Connectivity 8-Connectivity Mixed-Connectivity

ADJACENCY

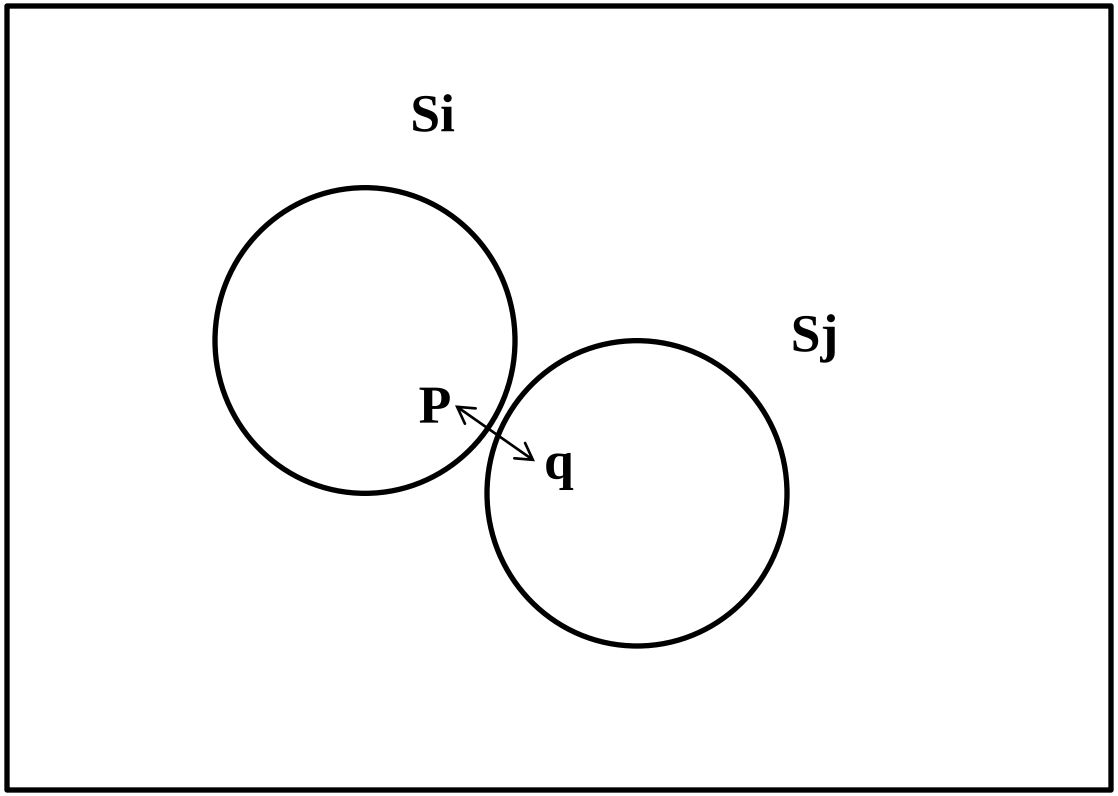
Two pixels p and q are adjacent if they are connected.
There are three different methods depending on type of connectivity Used.

- 4-adjacency
- 8-adjacency
- M-adjacency

ADJACENCY

Two image subsets S_i and S_j are adjacent if p Є S_i and q Є S_j such that p and q are connected.

Example



DISTANCE MEASURES

Take three Pixels

P = (x,y) , q=(s,t) and z=(u,v)

D is a distance function or metric if

$D(p,q) \geq 0$;

if p=q $D(p,q)=0$ $D(p,q)=D(p,q)$

$D(p,z) \leq D(p,q) + D(q,z)$

Euclidean distance

$$D_e(p,q) = [\ (x-s)^2 + (y-t)^2\ ]^{1/2}$$

Where

$P = (x,y)$, $q=(s,t)$

Set of points $S=\{\ q \mid D(p,q) \leq r\ \}$

are the points contained in a disk of radius r centered at p.

City block distance

D_4 distance or City Block (Manhattan) Distance.

$$D_4(p,q) = |x-s| + |y-t|$$

Points having City Block distance from p less than or equal to r from diamond centered at p.

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

Chess board distance

D_8 distance or Chess Board distance is defined as

$$\mathbf{D_8(p,q) = \max(|x-s|,|y-t|)}$$

$S = \{q \mid D_8(p,q) \leq r\}$ form a square centered at p.

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

Example 3.1 Let $V = \{0, 1\}$. Compute the D_e, D_4, D_8, and D_m distances between two pixels p and q. Let the pixel coordinates of p and q be (3, 0) and (2, 3), respectively, for the image shown in Fig. 3.10.

Find the distance measures.

$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 \\
2 & 1 & 1 & 1 & 1\,(q) \\
3 & 1 & 1 & 1 & 1 \\
 & (p) & & &
\end{array}
$$

Fig. 3.10 Sample image

Solution The Euclidean distance is

$$D_e = \sqrt{(x-s)^2 + (y-t)^2} = \sqrt{(3-2)^2 + (0-3)^2}$$

$$= \sqrt{1+9} = \sqrt{10}$$

$$D_4 = |x-s| + |y-t| = |3-2| + |0-3|$$

$$= 1 + 3 = 4$$

$$D_8 = \max(|x-s|, |y-t|) = \max(|3-2|, |0-3|)$$

$$= \max(1, 3) = 3$$

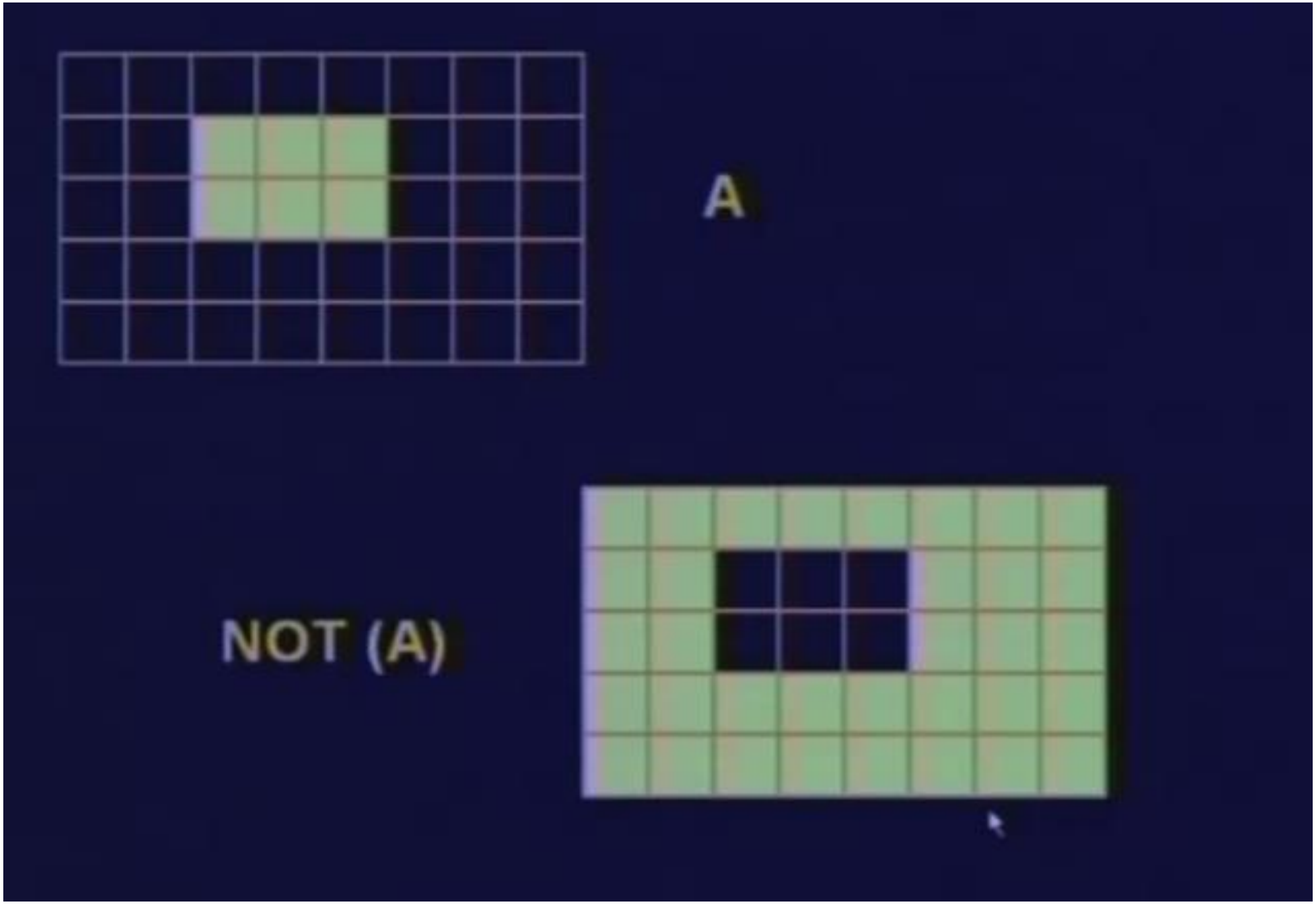
Arithmetic and logical operation

Following Arithmetic/Logical operations between two pixels p and q are used extensively

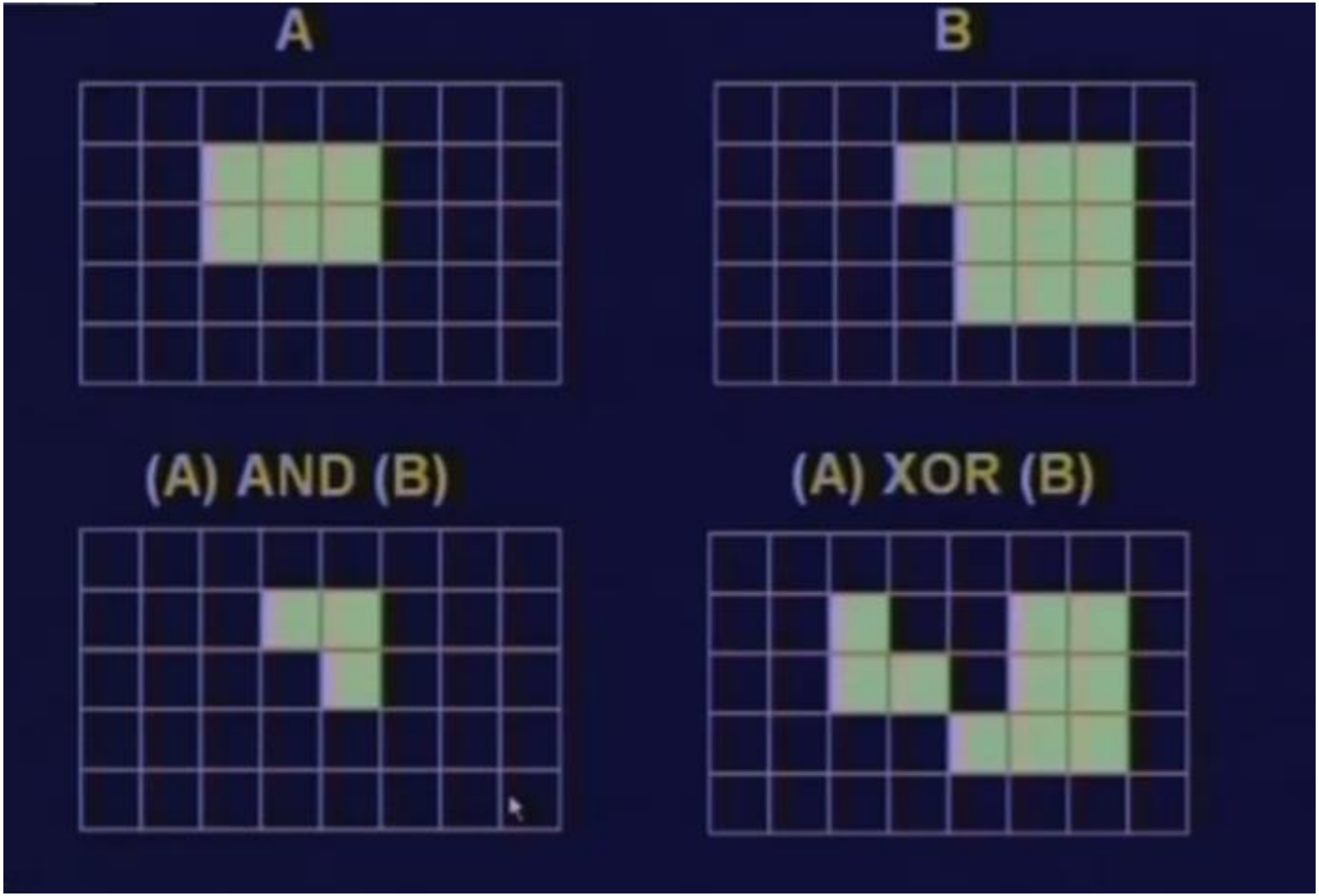
Arithmetic	Logical
p+q	p.q
p-q	p+q
p*q	p'
p%q	

Logical operations apply to binary images Only => Usually pixel by

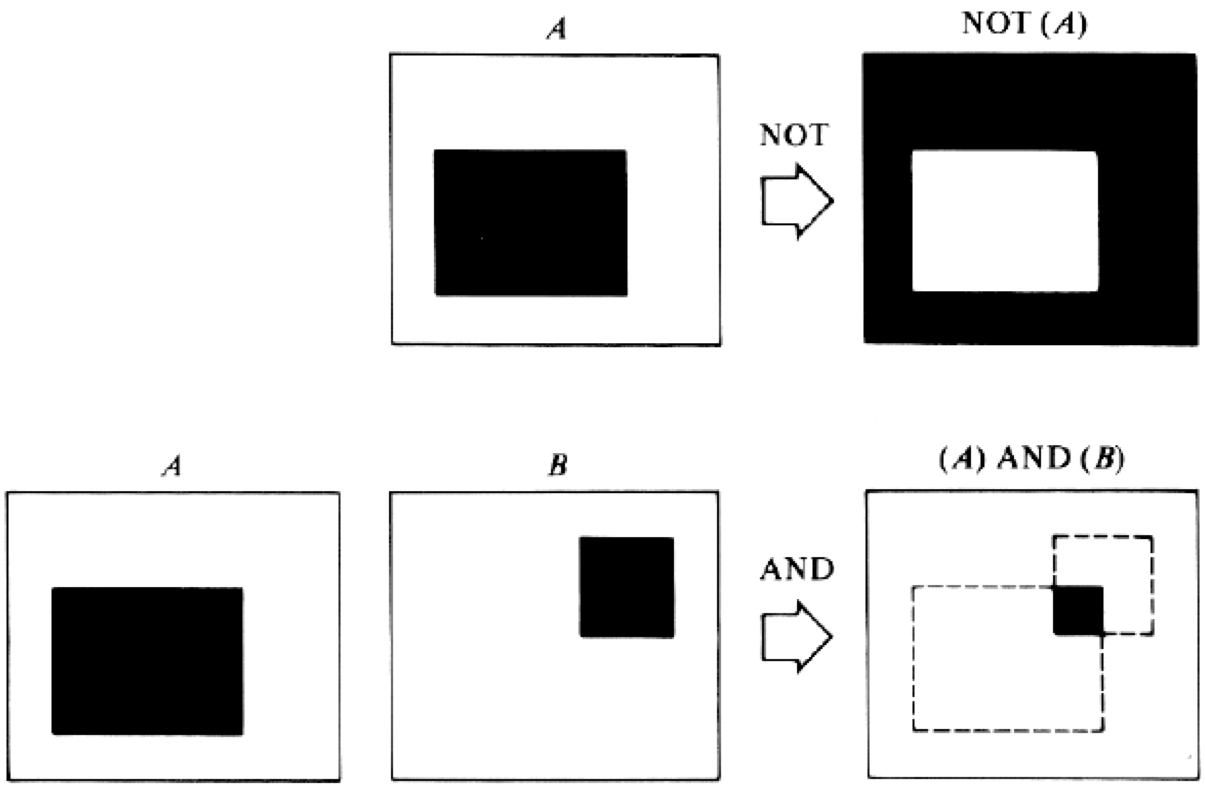
Arithmetic and logical operation



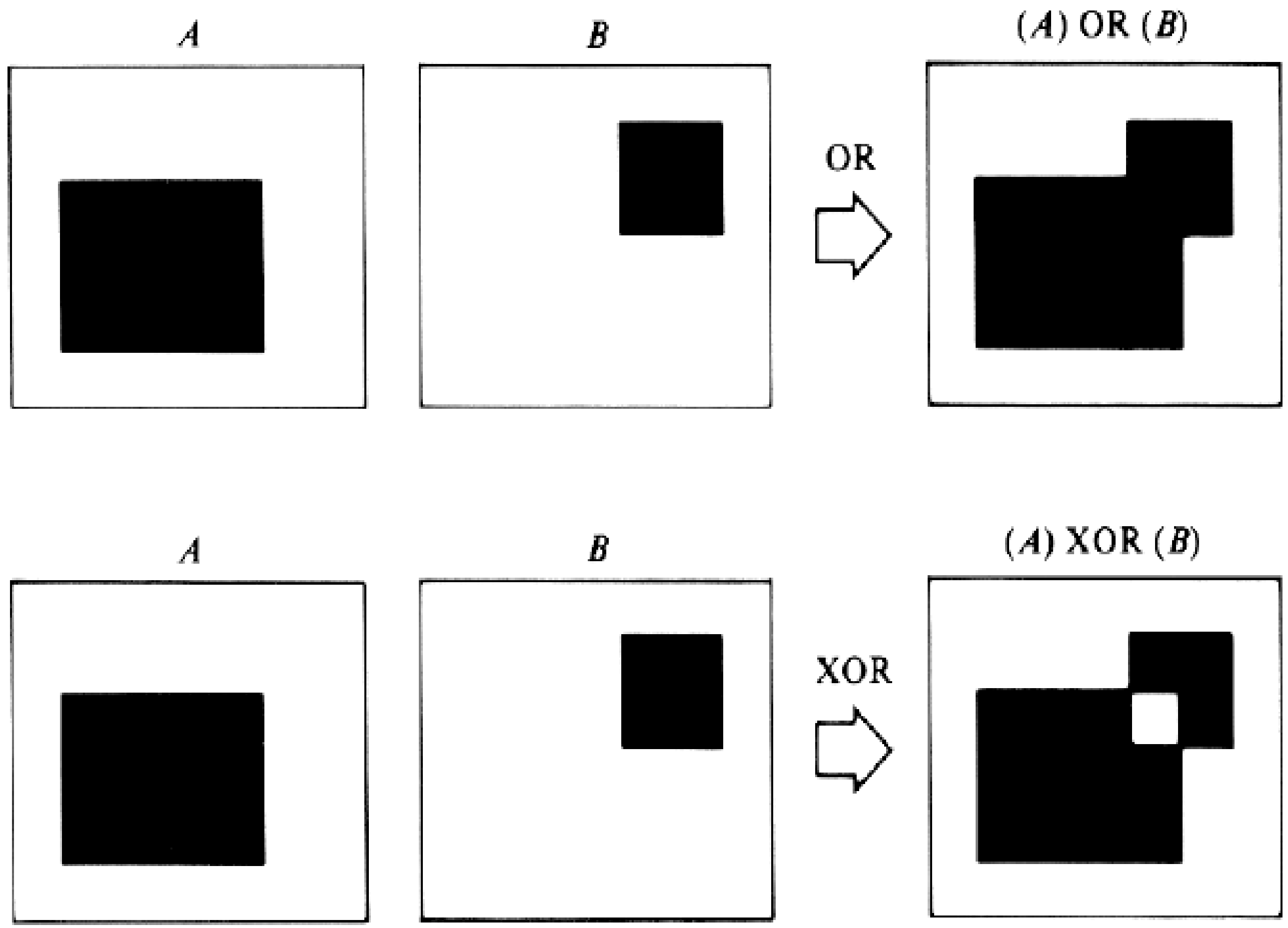
Arithmetic and logical operation



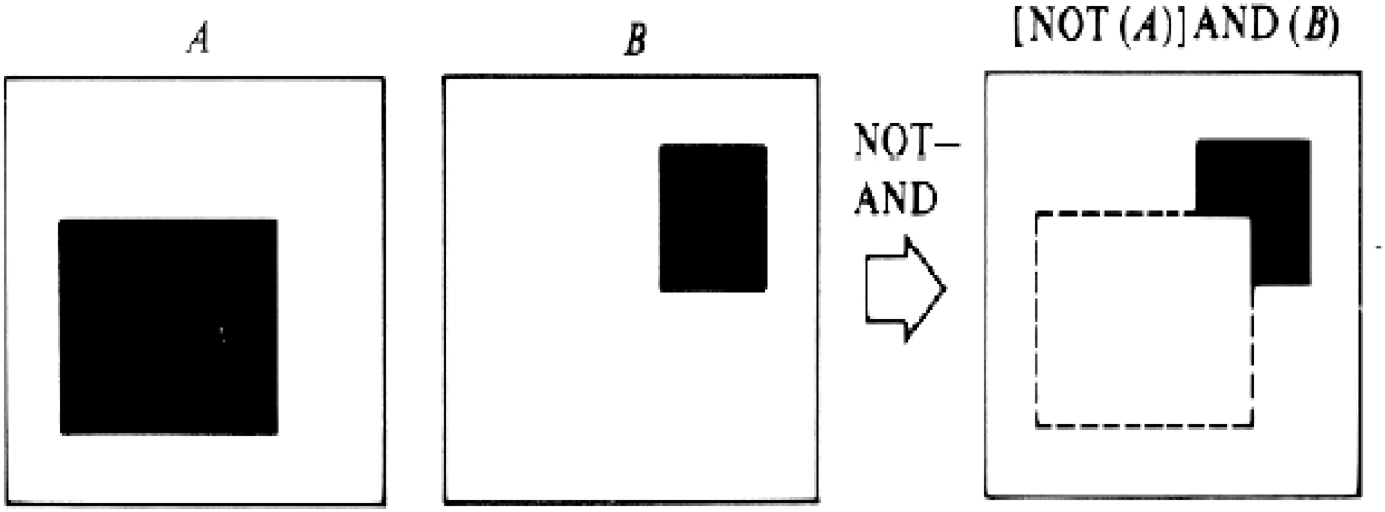
ARITHMETIC AND LOGICAL OPERATION



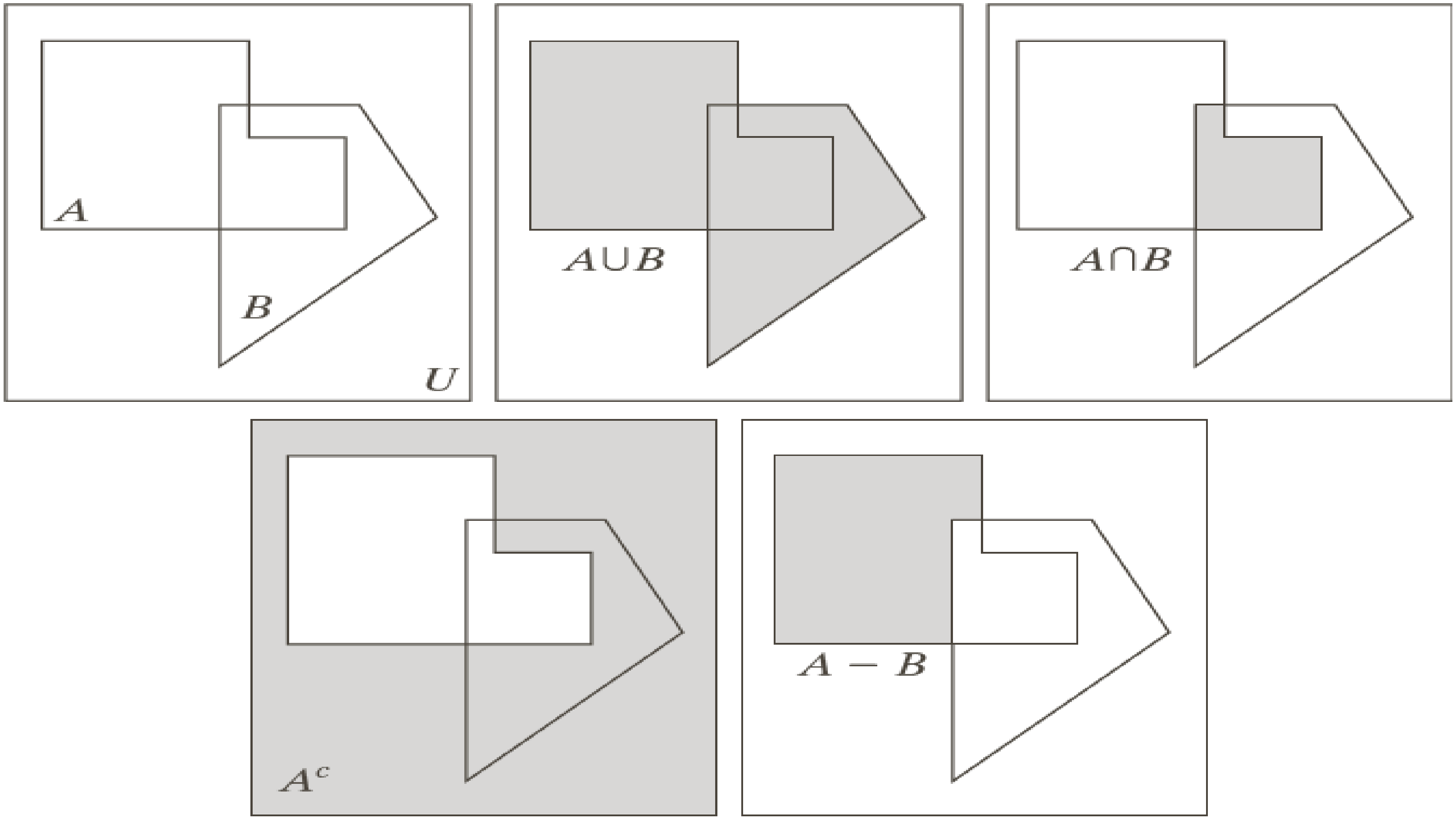
Arithmetic and logical operation



ARITHMETIC AND LOGICAL OPERATION



ARITHMETIC AND LOGICAL OPERATION



NEIGHBOURHOOD OPERATION

The value assigned to a pixel is a function of its gray label and the gray labels of its neighbors.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

$$Z = 1/9\,(Z_1+Z_2+Z_3+\ldots\ldots+Z_9) = \text{Average}$$

TEMPLATE

More general form

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

$$Z = W_1Z_1 + W_2Z_2 + \ldots\ldots\ldots + W_9Z_9$$

$$= \sum_{i=1}^{9} W_iZ_i$$

Same as averaging if $W_i = 1/9$

NEIGHBOURHOOD OPERATION

Various important operations can be implemented by proper selection of Coefficients Wi

- **Noise Filtering-** It is used to remove noise from image.
- **Thinning** – It is used to remove selected foreground pixels from binary images.
- **Edge Detection-** It is used to detect edge of an Image

Etc.

LINEAR AND NONLINEAR OPERATIONS

- Let H be an operator whose input and output are images. H is said to be a *linear* operator if, for any two images f and g and any two scalars a and b,

$$\mathbf{H(af + bg) = aH(f) + bH(g).}$$

- Linear operations are exceptionally important in image processing because they are based on a significant body of well-understood theoretical and practical results.
- Although nonlinear operations sometimes offer better performance, they are not always predictable, and for the most part are not well understood theoretically.

Thanks

Image Geometry

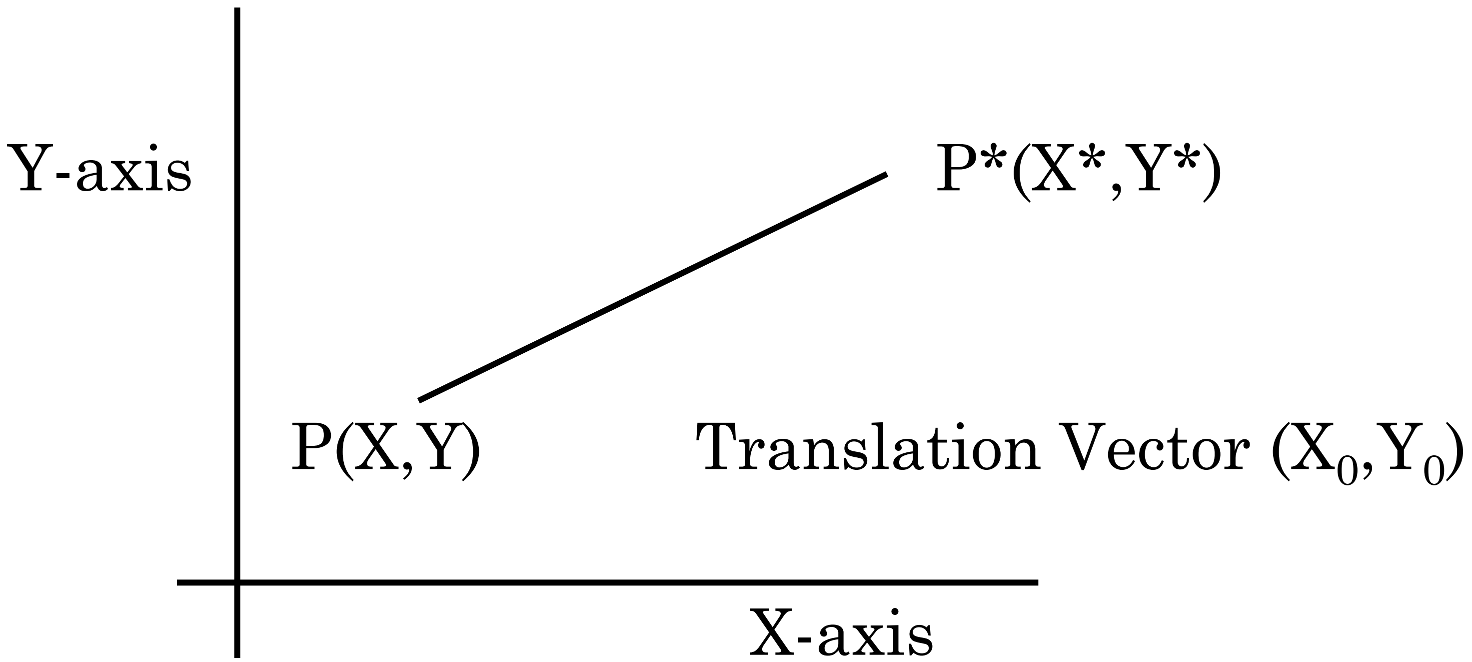
By,

Prof (Dr) P P Ghadekar

OUTLINE

- On completion student will be able to understand
- Basic transformation-Translation, Rotation and Scaling in both 2-D and 3-D Image .
- Inverse transformation.

WHAT IS TRANSLATION OPERATION



Then, $X^* = X + X_0$

$Y^* = Y + Y_0$

Image Translation in 2-D

$$\begin{bmatrix} X^* \\ Y^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

Above matrix equation can be represented in Single matrix form

$$\begin{bmatrix} X^* \\ Y^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & X \\ 0 & 1 & Y \end{bmatrix} * \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix}$$

Image Translation in 2-D

- Translation matrix in symmetric form

$$\begin{bmatrix} X^* \\ Y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_0 \\ 0 & 1 & Y_0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Above expression also called as Unified expression

Image translation in 3-D

- Suppose that the task is to translate a point with coordinates (X,Y,Z) to new location by using displacements (X_0,Y_0,Z_0). The translation is easily accomplished by using the equations.

$$X^*=X+X_0$$

$$Y^*=Y+Y_0$$

$$Z^*=Z+Z_0$$

Where X^*,Y^*, Z^* are the coordinates of the new location. Above equation in matrix form is -

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \text{-----(I)}$$

Image translation

- Square matric representation of the matrix is

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

--------(II)

In terms of the values of X^*,Y^*, Z^* are equivalent in equations (I) and (II).

Image translation

We can also used the unified matrix representation as given below

$$\mathbf{V^*=TV}$$

Where T is 4*4 transformation matrix, V is column vector containing the original coordinates and V^* is the column vector whose components are transformed coordinates.

$$\mathbf{v} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \mathbf{v}^* = \begin{bmatrix} X^* \\ Y^* \\ Z^* \\ 1 \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So that above equation becomes $\mathbf{V^*=TV}$

Image scaling

- Scaling by factor S_X, S_Y and S_Z along the X, Y, and Z axes is given by transformation matrix

$$\mathbf{S} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

IMAGE ROTATION

- To rotate a point about another arbitrary point in space requires three transformations as given below

I. Translates the arbitrary point to the origin.

II. Performs the rotation

III. Translates the point back to its original position.

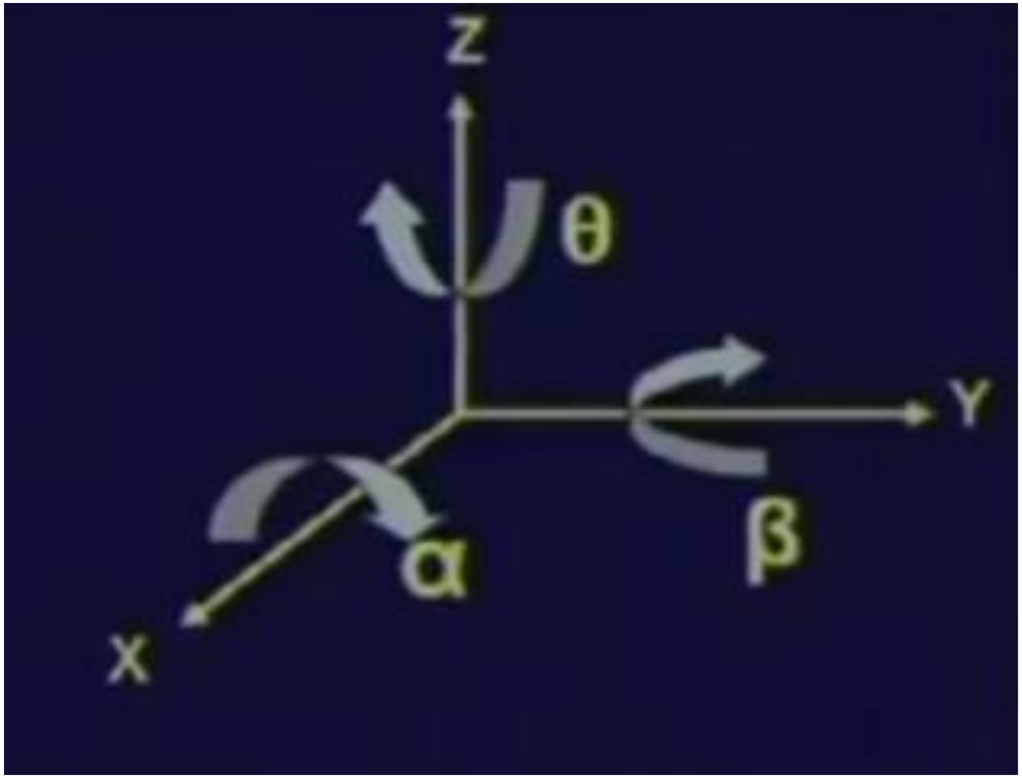


Image rotation

- Rotation of a point about Z-axis by angle θ is achieved by using the transformation matrix given below.

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The rotation angle θ is measured clockwise when looking at the origin from a point on the Z-axis.
- This transformation affects values of X and Y coordinates.

Image rotation

- Rotation of a point about the X-axis by an angle α is performed by using the following transformation.

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

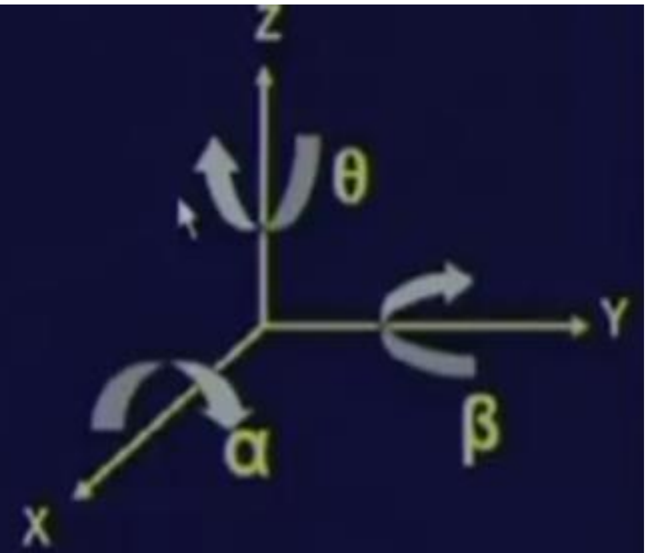
IMAGE ROTATION

- Rotation of a point about the Y-axis by an angle β is performed by using the following transformation.

$$\mathbf{R}_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

IMAGE ROTATION

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$R_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

INVERSE TRANSFORMATION

- Inverse translation matrix is given below.

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Similarly the inverse rotation matrix is given below

$$\mathbf{R}_\theta^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) & 0 & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

INVERSE TRANSFORMATION

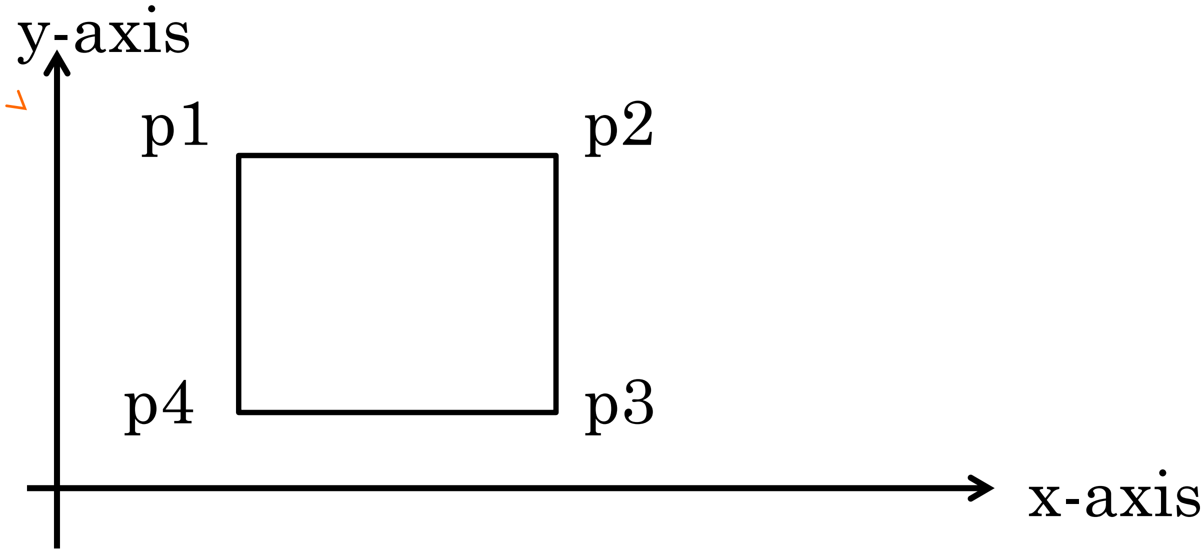
Can be Obtained by Observations

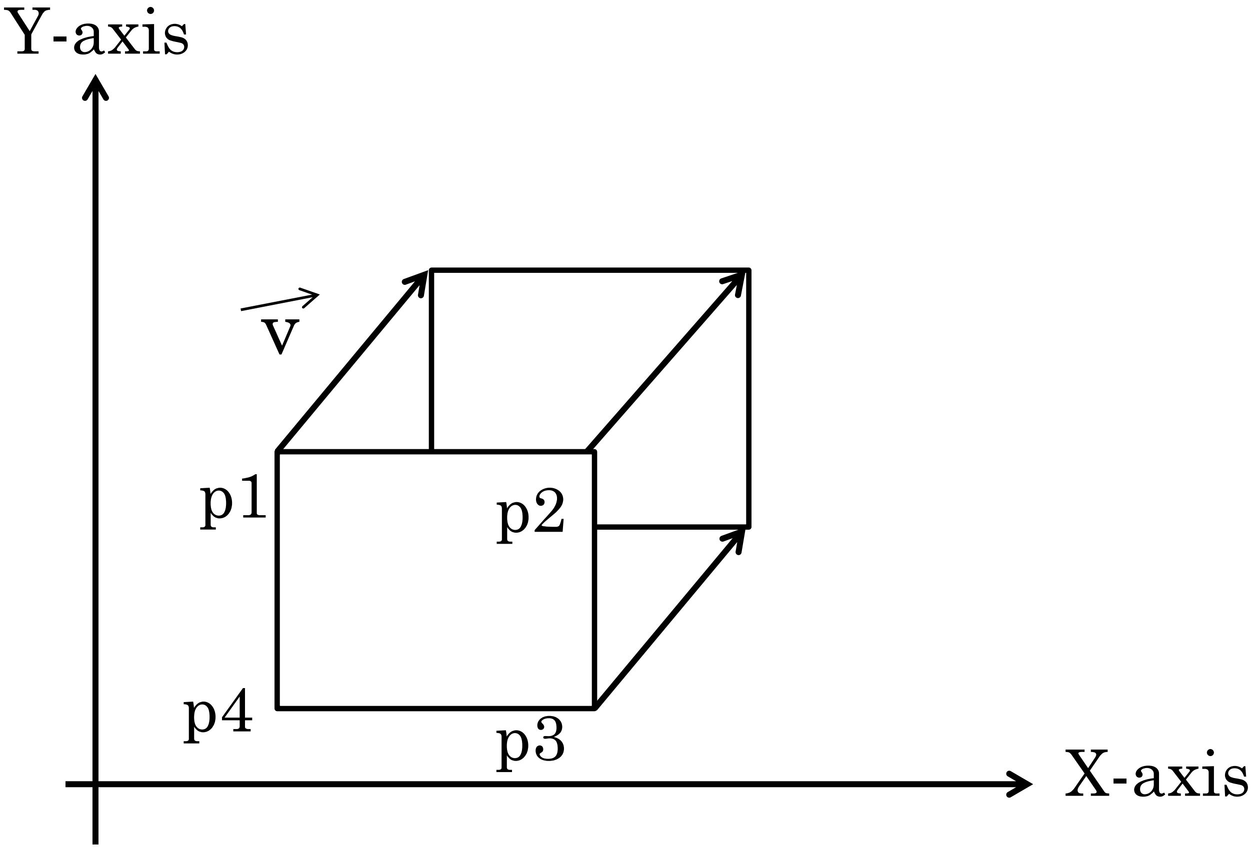
$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R_\theta^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) & 0 & 0 \\ -\sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation for a set of points

- For a set of m-points construct a matrix- V of dimension 4 x m.

The Transformation $V^* = TV$




Y-axis
v
p1
p2
p4
p3
X-axis

Thanks

Image Processing & Computer Vision

By,
Prof (Dr.) P P Ghadekar

Outline

- ❖ Basic Concepts of Image Processing
- ❖ Key Stages of Image Processing
- ❖ Image Formation
- ❖ Image Processing Applications & Research Areas etc.

Why Image Processing

- To Extract useful information/intelligence from the image.
- To study nature and characteristics of an image.
- To analyze the image.
- To study various attributes of the image.
- To improve quality of an image for human interpretation.
- To process an image data for storage, transmission and representation for autonomous machine perception.

IT-Image Processing & CV

❖ Prerequisite:

- Knowledge of Different types of Signals, Linear Algebra, Probability and Statistics

❖ Course Outcomes:

- The student will be able to

1. Apply lossless and Lossy compression techniques for image compression.
2. Explore pre-processing algorithms to acquire images
3. Use Wavelet transforms to analyze and modify image
4. Extract features from Images and do analysis of Images
5. Apply Segmentation techniques on Images
6. Make use of Computer Vision algorithms to solve real-world problems.

Image Processing & Computer Vision-Syllabus

❖ Section-I

- ➢ Introduction to Image Processing
- ➢ Image Enhancement
- ➢ Image compression and its need

❖ Section-II

- ➢ Introduction to Computer Vision
- ➢ Shape Representation and Segmentation
- ➢ Object recognition, Camera Calibration and 3-D Images
- ➢ Machine Learning for Image classification

Image Processing & Computer Vision Syllabus

IP & CV-Books

❖ Text Books

- Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.
- S. Jayaraman, S Esakkirajan, & T Veerakumar, "Digital Image Processing," Tata McGraw Hill Education, ISBN(13) 9780070144798.
- Anil K. Jain, "Fundamentals of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.
- Richard Szeliski, "Computer Vision: Algorithms and Applications (CVAA)", Springer, 2010.
- E. R. Davies, "Computer & Machine Vision," 4th Edition, Academic Press, 2012.
- Simon J. D. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012.

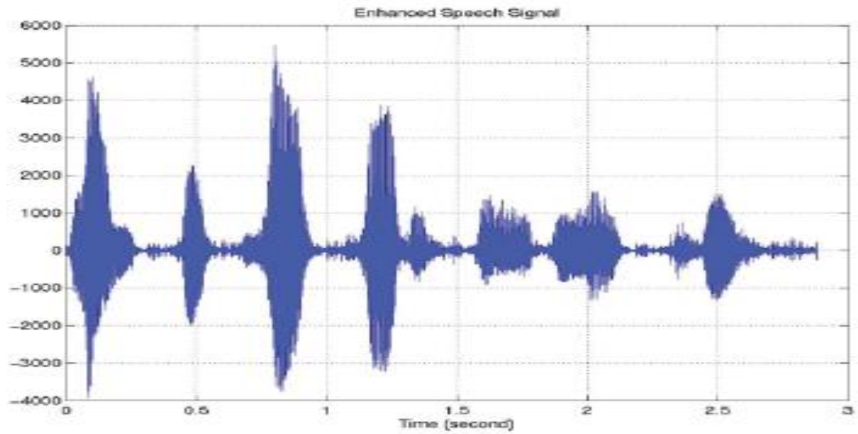
Background

❖ What is a Signal?

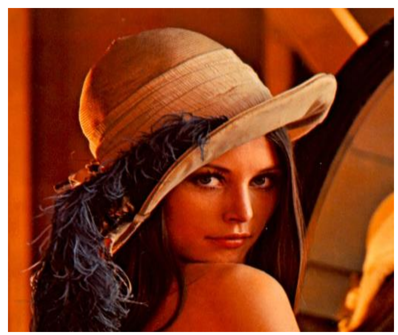
- A function of one or more independent variables such as time, frequency, distance, position, temperature, pressure, etc.
- A signal carries information.
- Examples: speech, music, image and video.
- A signal can be a function of one, two or N independent variables.
  - Speech is a 1-D signal as a function of time.
  - An image is a 2-D signal as a function of space.
  - Video is a 3-D signal as a function of space and time.

More Examples

1-D Signal
Speech Signal



2-D Signal
Image Signal



3-D Signal
Video Signal

time

Types of Signals

❖ Analog Signal

Signals that are continuous in both the dependent and independent variable (e.g., amplitude and time). Most environmental signals are continuous-time signals.

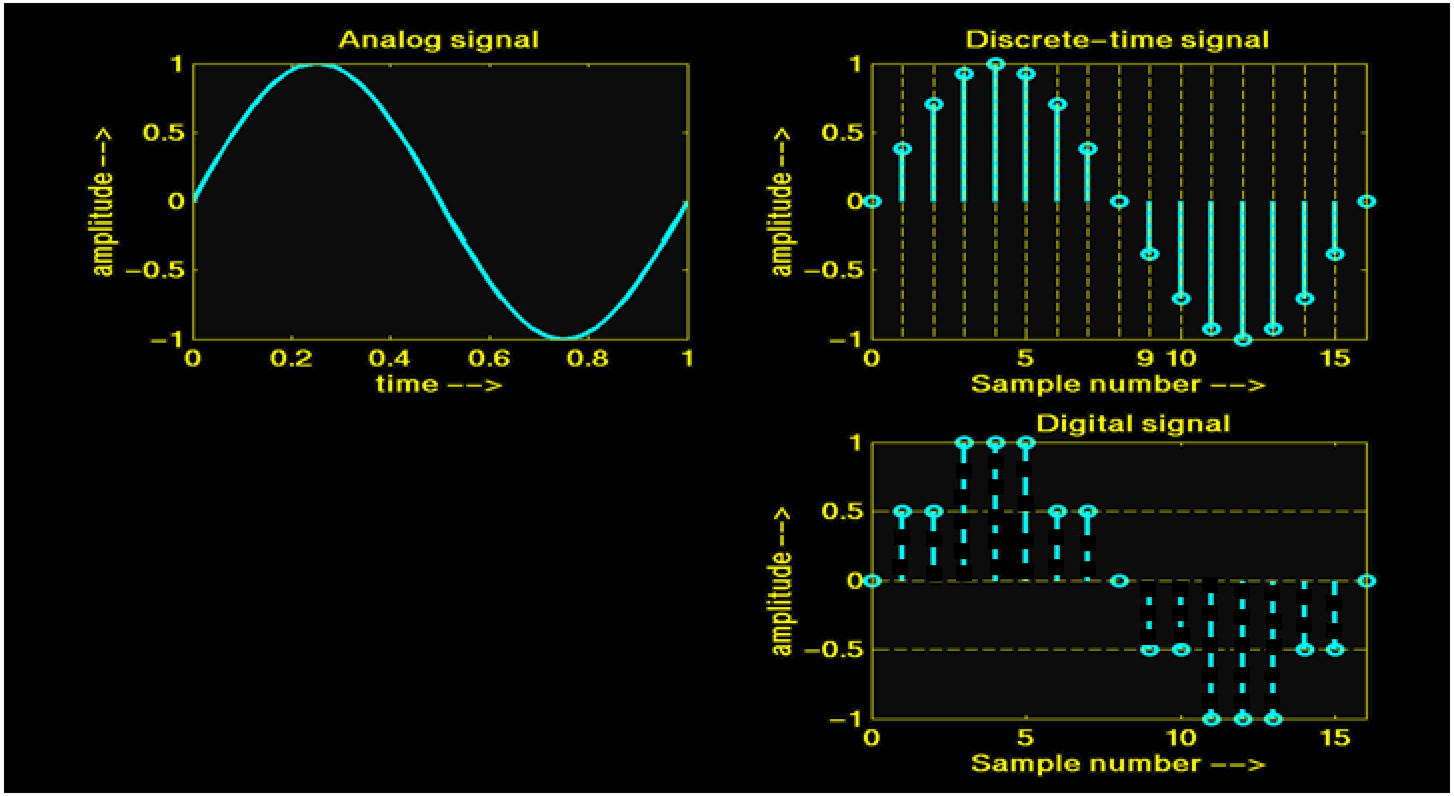
❖ Discrete Sequences (Discrete-Time Signals)

Signals that are continuous in the dependent variable (e.g., amplitude) but discrete in the independent variable (e.g., time). They are typically associated with sampling of continuous-time signals.

Types of Signals (cont.)

Digital Signal

Signals that are discrete in both the dependent and independent variable (e.g., amplitude and time) are digital signals. These are created by quantizing and sampling continuous-time signals or as data signals (e.g., stock market price fluctuations).



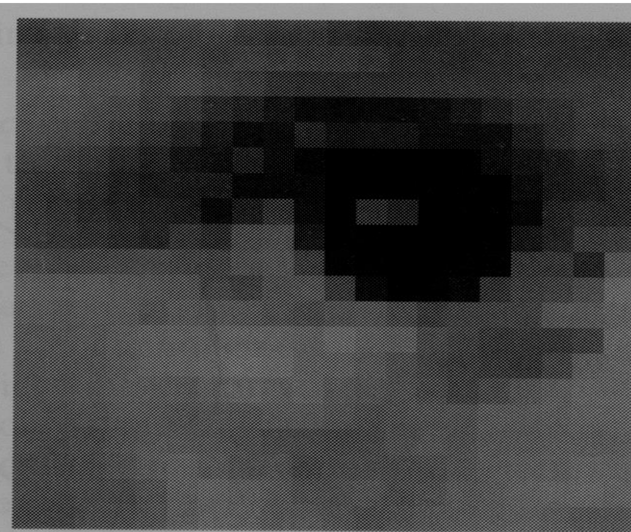
Introduction to Image Processing

"One picture is worth more than ten thousand words"

Image

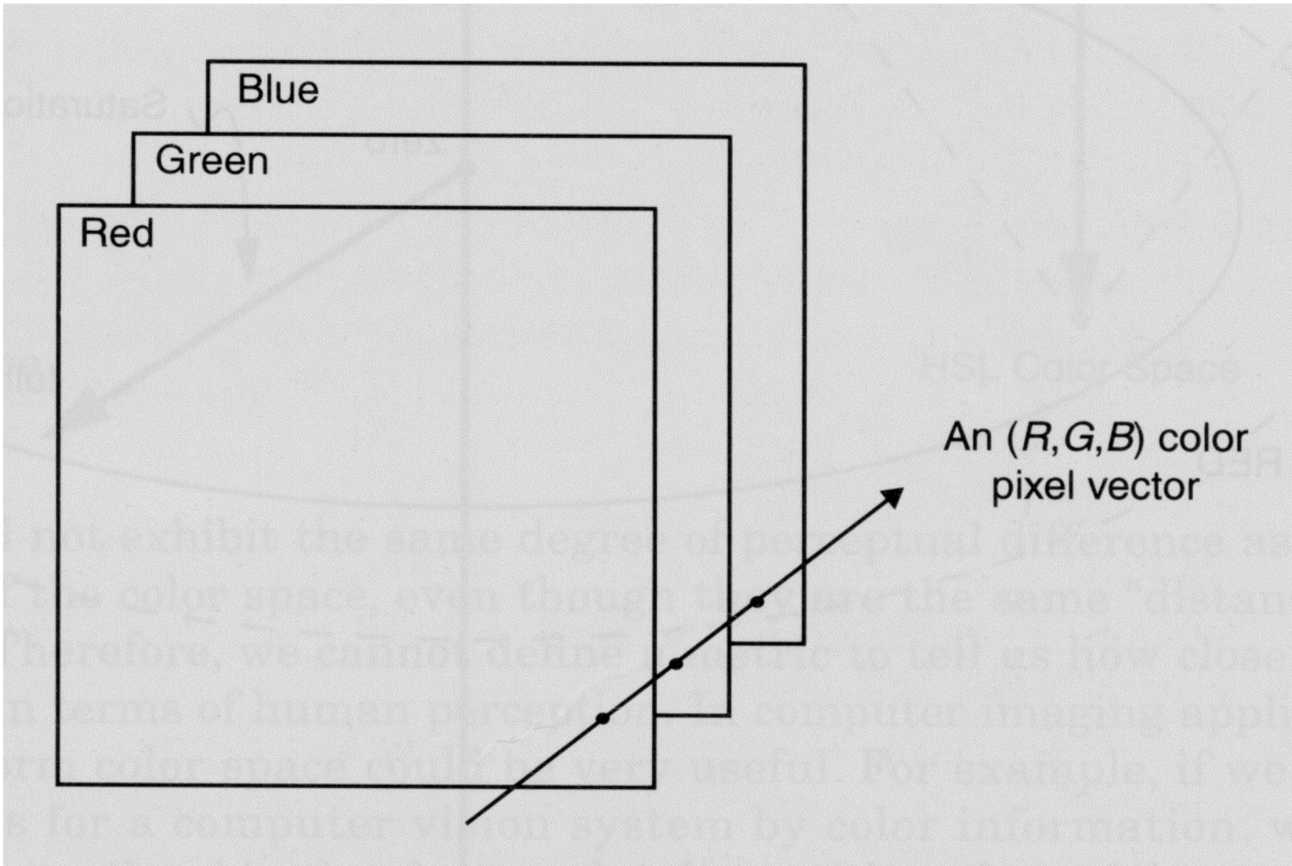
- **What is an image ?**
  - An image may be defined as a two-dimensional function, f(x,y) , where x and y are spatial (plane) coordinates , and the magnitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point.
  - An image is a 2-d rectilinear array of pixels

How are Images represented in the Computer

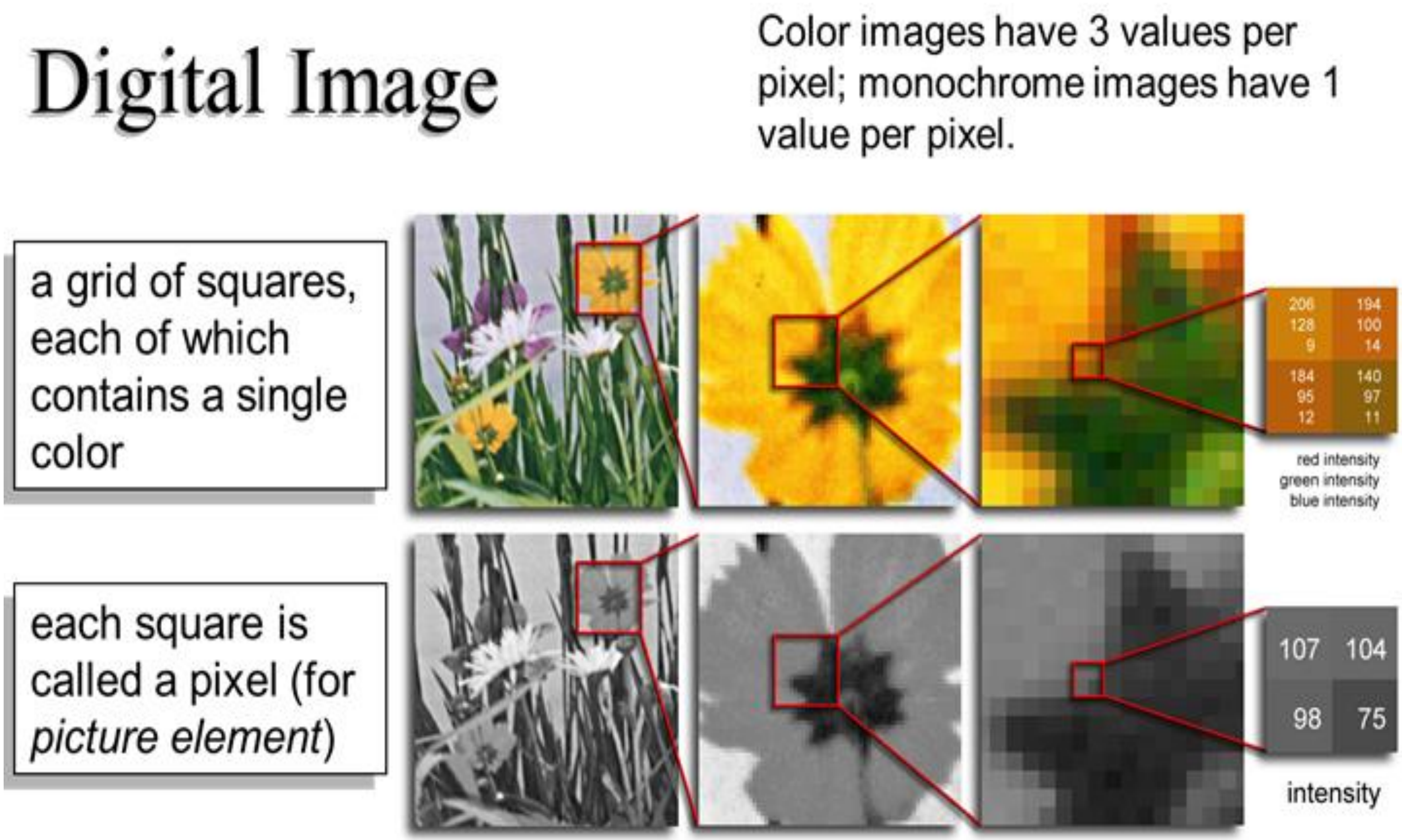


117	125	133	127	130	130	133	121	116	115	100	91	93	94	99	103	112	105	109	106
134	133	138	138	132	134	130	133	128	123	121	113	106	102	99	106	113	109	109	113
146	147	138	140	125	134	124	115	102	96	93	94	99	96	99	100	103	110	109	110
144	141	136	130	120	108	88	74	53	37	31	37	35	39	53	79	93	100	109	116
139	136	129	119	102	85	58	31	41	77	51	53	53	33	37	41	69	94	105	108
132	127	117	102	87	57	49	77	42	28	17	15	13	13	17	41	53	69	88	100
124	120	108	94	72	74	72	31	35	31	15	13	15	11	15	13	46	75	83	96
125	115	102	93	88	82	42	79	113	41	19	100	82	11	11	17	31	91	99	100
124	116	109	99	91	113	99	140	144	57	20	20	15	11	15	17	63	87	119	124
136	133	133	135	138	133	132	144	150	120	24	17	15	15	17	20	115	113	88	150
158	157	157	154	149	145	133	127	146	150	116	35	20	19	28	105	124	128	141	171
155	154	156	155	146	155	154	154	147	139	148	150	138	120	128	129	130	151	156	165
150	151	154	162	166	167	169	174	172	167	177	166	164	140	134	120	121	120	127	172
145	149	151	157	165	169	173	179	176	166	166	157	145	136	129	124	120	136	163	168
144	148	153	160	159	158	165	172	165	169	157	151	149	141	130	140	151	162	169	167
144	141	147	155	154	149	156	151	157	157	151	144	147	147	149	159	158	159	166	165
139	140	140	150	153	151	150	146	140	139	138	140	145	151	149	156	156	162	162	161
136	134	138	146	156	164	153	146	145	136	139	139	140	141	149	157	159	161	169	166
136	133	136	135	144	159	168	159	151	142	141	145	139	146	153	156	164	167	172	168
133	129	140	142	146	159	167	165	154	151	146	141	147	154	156	160	161	157	153	154

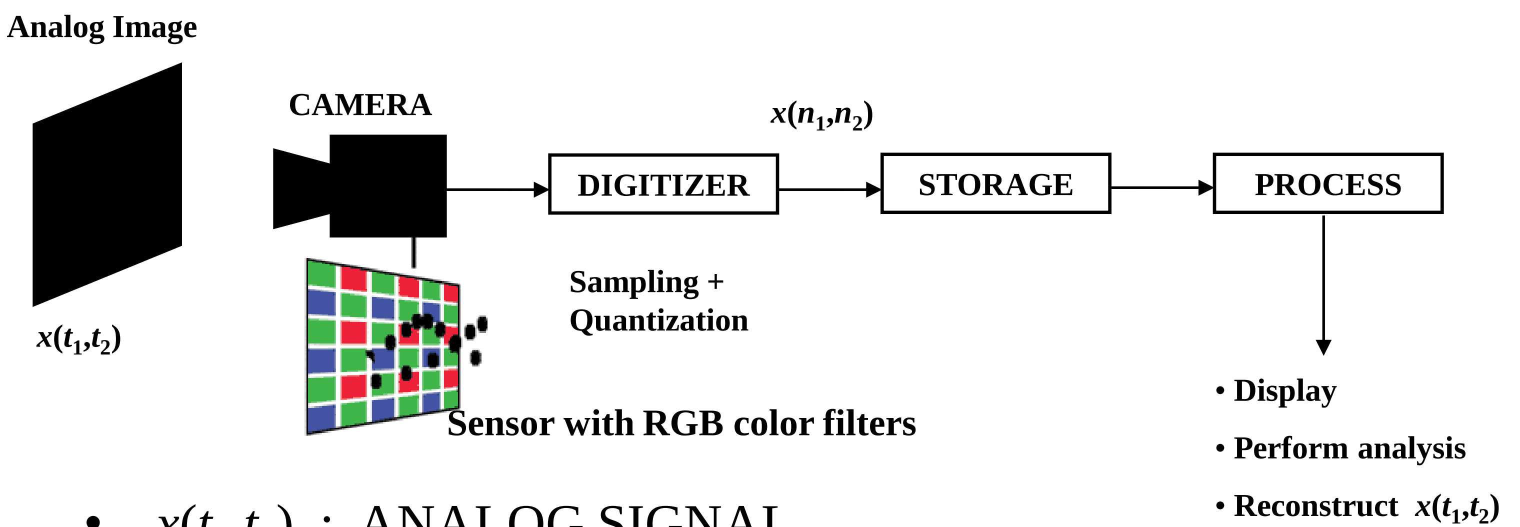
Color Images



Digital Color Image



What is Image Processing?



- $x(t_1,t_2)$: ANALOG SIGNAL

  x : real value

  (t_1,t_2) : pair of real continuous space (time) variables

- $x(n_1,n_2)$: DISCRETE SIGNAL (DIGITAL)

  x : discrete (quantized) real or integer value

  (n_1,n_2) : pair of integer indices

Examples

- *Sampled Black & White Photograph:* $x(n_1,n_2)$

  $x\ (n_1,n_2)$ scalar indicating pixel intensity at location (n_1,n_2)

  For example: x = 0 Black

  x = 1 White

- Sampled color video/TV signal

  $x_R(n_1, n_2, n_3)$

  $x_G(n_1, n_2, n_3)$

  $x_B(n_1, n_2, n_3)$

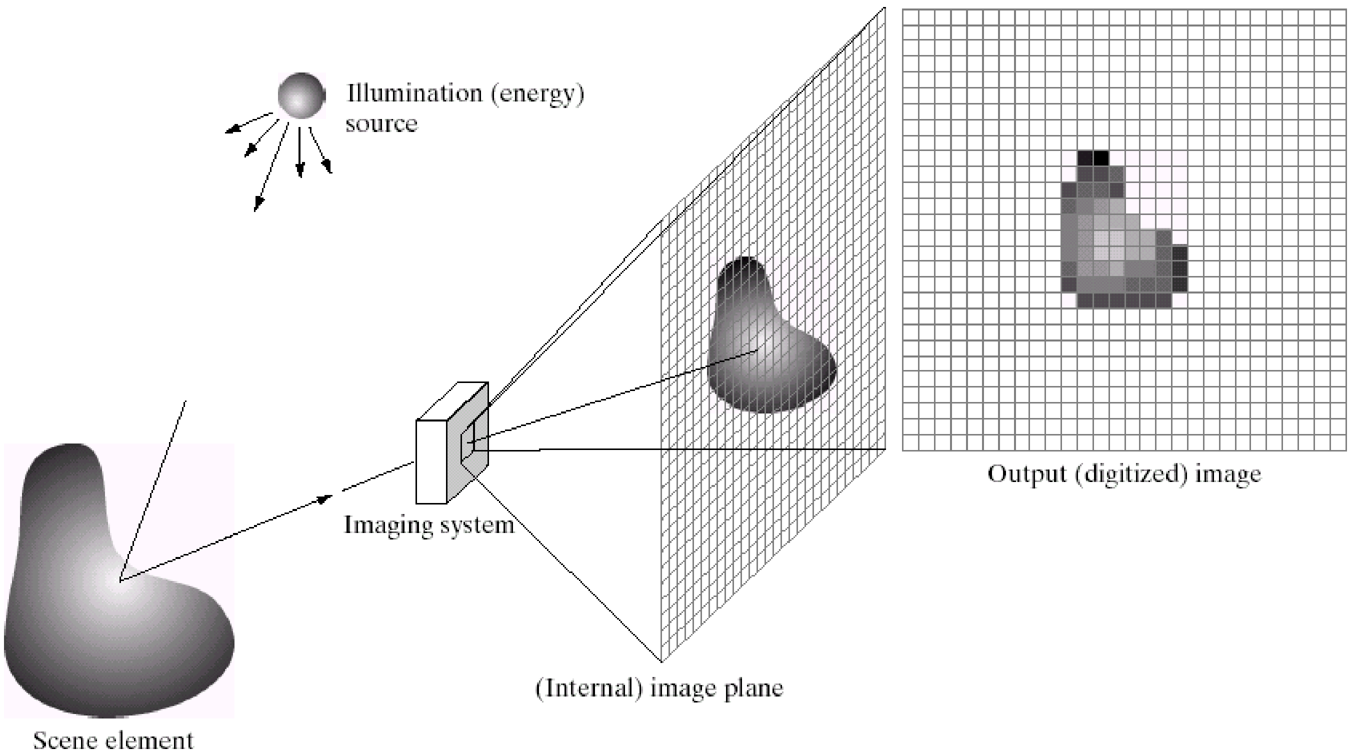
Examples

Gray Scale Image

Colour Image

What is a Digital Image?

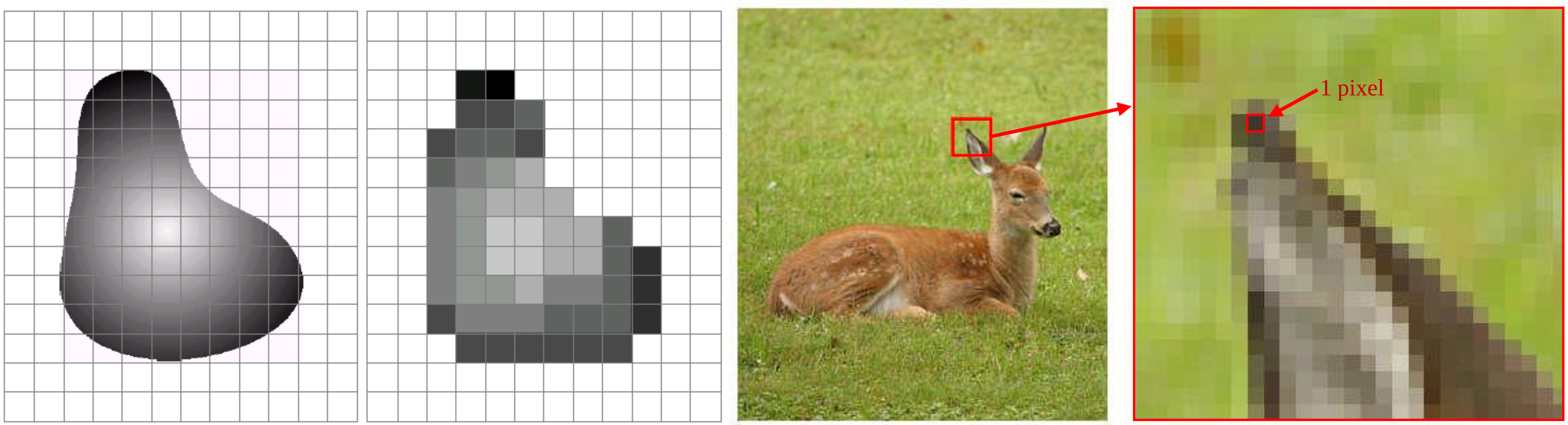
•A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements, pel, or pixels



What is a Digital Image? (cont…)

•Pixel values typically represent gray levels, colours, heights, opacities etc

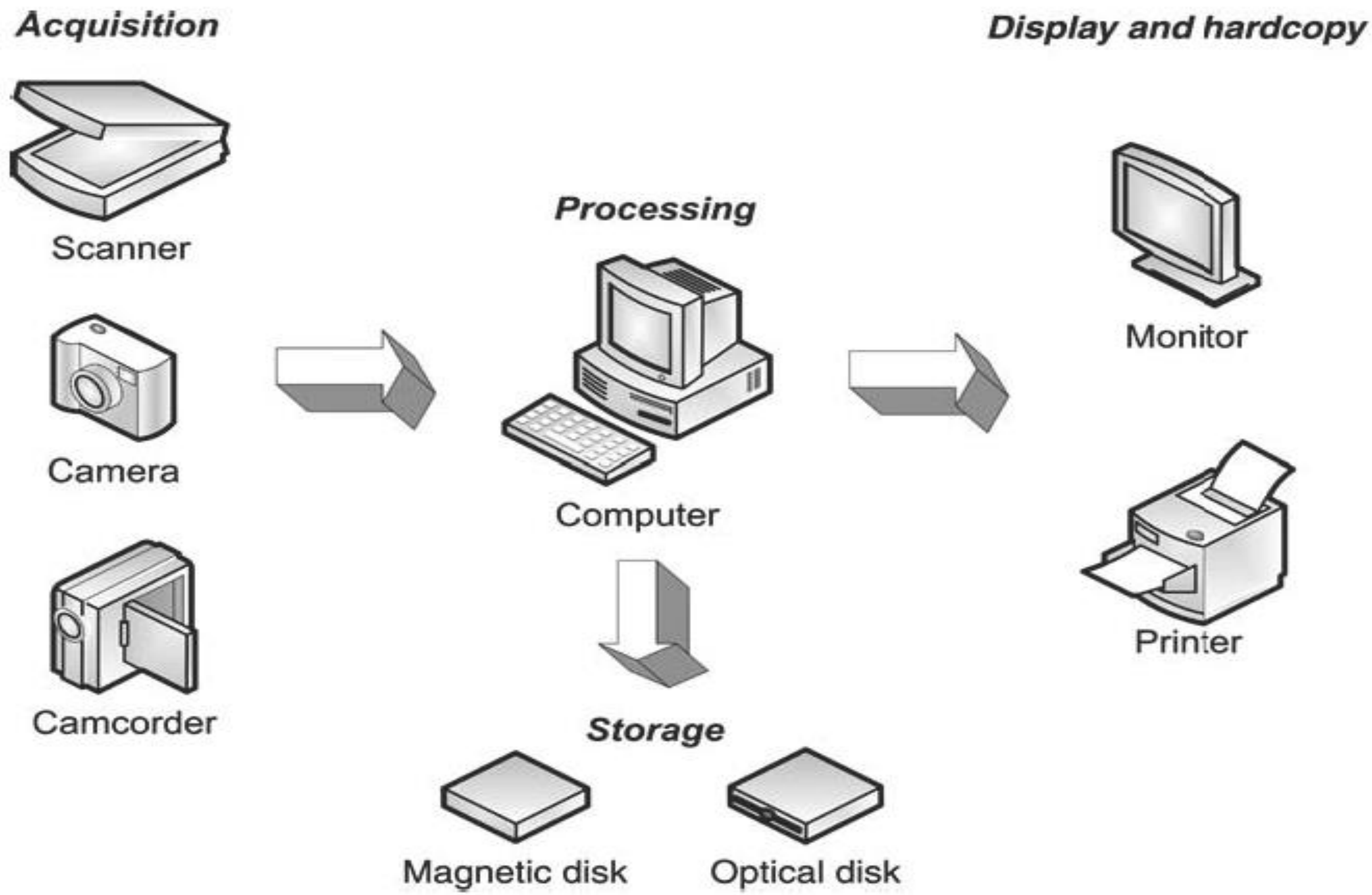
•Digitization implies that a digital image is an *approximation* of a real scene



What is Digital Image Processing?

- Digital image processing focuses on two major tasks
  - Improvement of pictorial information for human interpretation.
  - Processing of image data for storage, transmission and representation for autonomous machine perception.

Digital Image Processing System



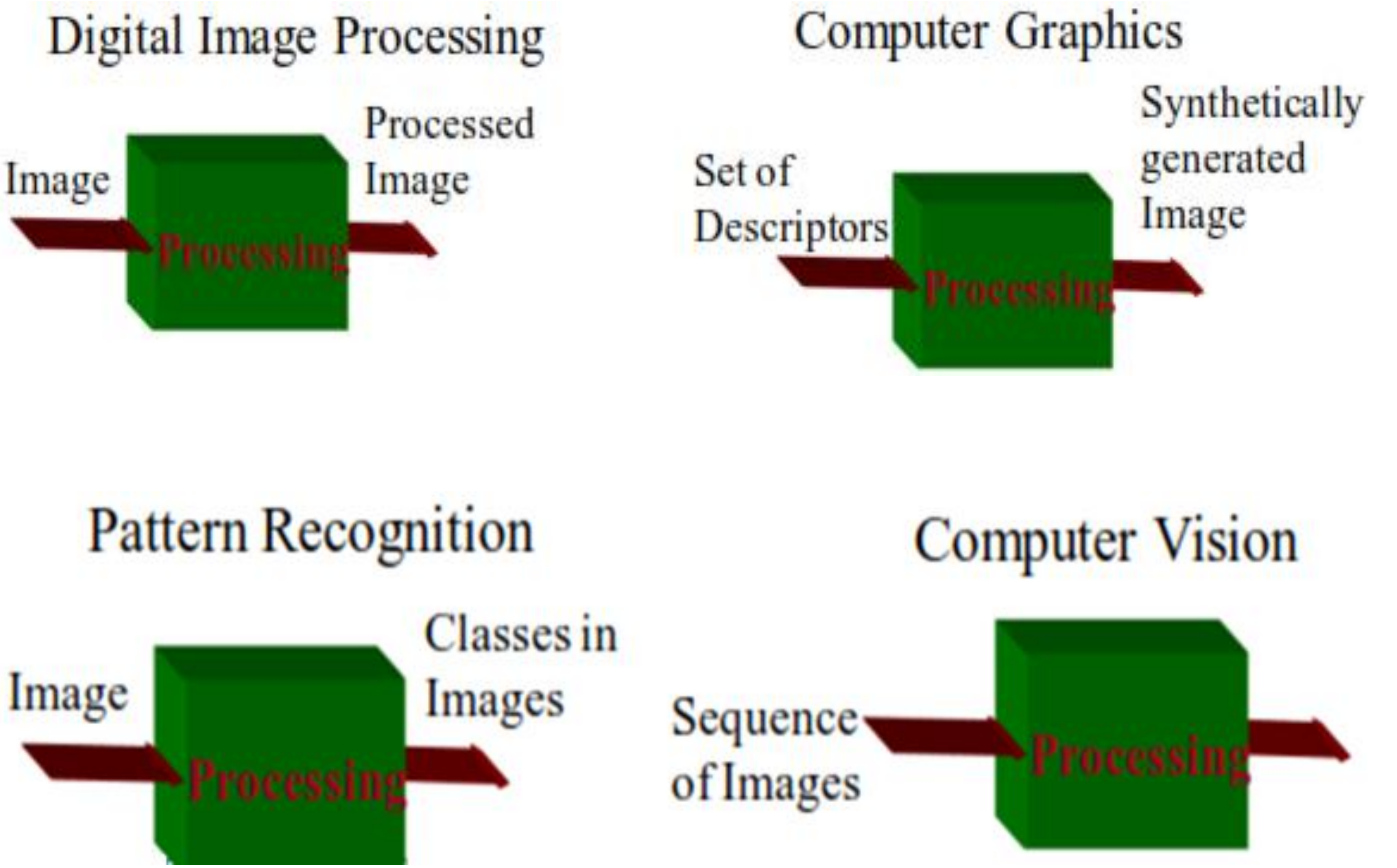
Digital Image Processing

❖The range from image processing to computer vision can be broken up into low-, mid- and high-level processes

Low Level Process	Mid-Level Process	High Level Process
Input: Image **Output:** Image **Examples:** Noise removal, image sharpening, Contrast enhancement etc.	**Input:** Image **Output:** Attributes **Examples:** Object recognition, segmentation, edges etc.	**Input:** Attributes **Output:** Understanding **Examples:** Scene understanding, autonomous navigation Etc.

Analogy-Character Recognition System

Four Areas


Digital Image Processing
Image
Processing
Processed Image
Computer Graphics
Set of Descriptors
Processing
Synthetically generated Image
Pattern Recognition
Image
Processing
Classes in Images
Computer Vision
Sequence of Images
Processing

History of Digital Image Processing

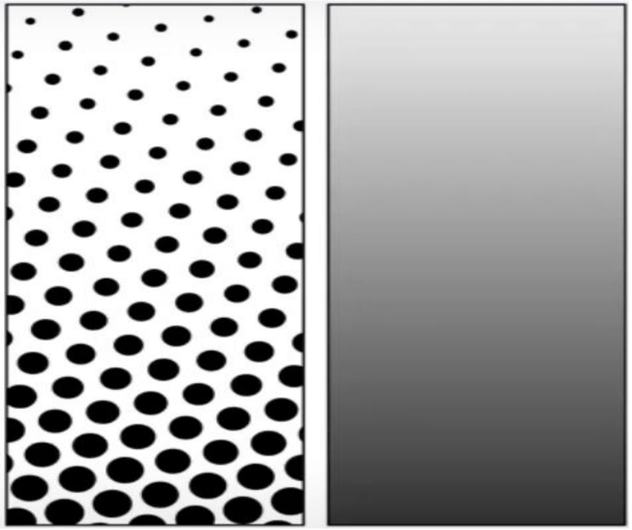
❖**Early 1920s:** One of the first applications of digital imaging was in the news- paper industry

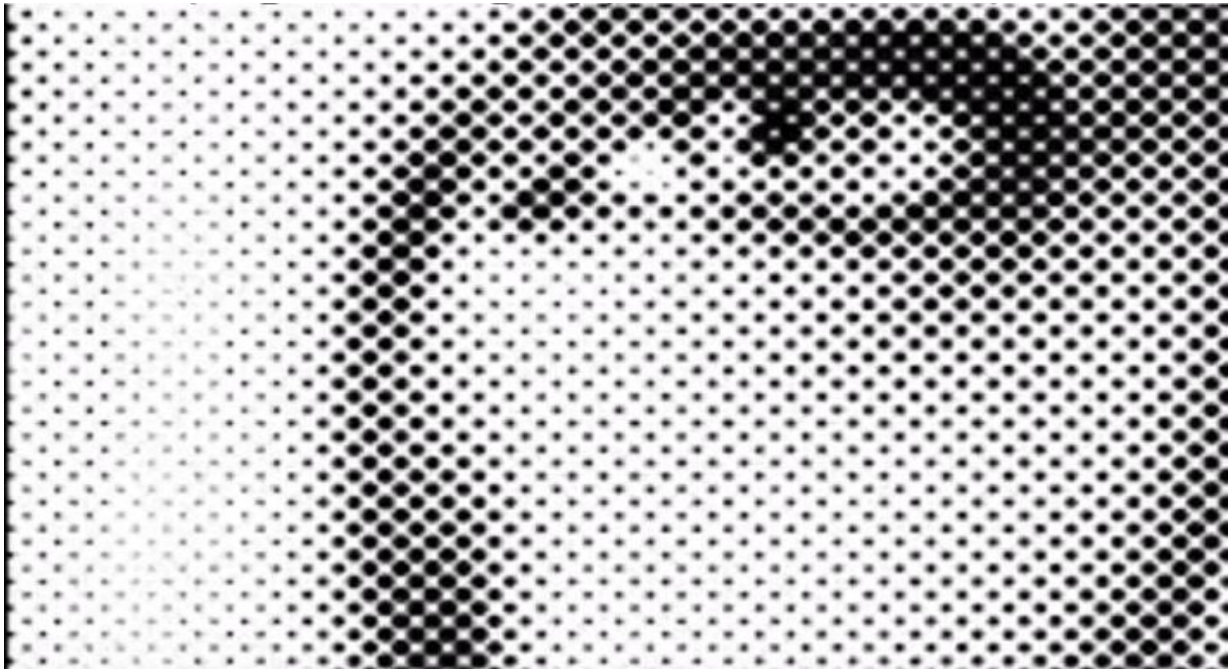
The **Bartlane cable picture** transmission service.

- Images were transferred by **Submarine cable** between London and New York.
- Pictures were coded for cable transfer and reconstructed at the receiving end on a **telegraph printer**

Early digital image

History of Digital Image Processing



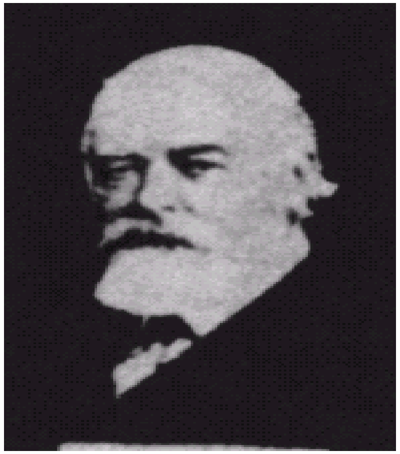


History of DIP (cont…)

•**1921: Improvements to the Bartlane system** resulted in higher quality images

- ➢ New reproduction processes based on photographic techniques
- ➢ **Increased number of tones** in reproduced images

Early 15 tone digital image

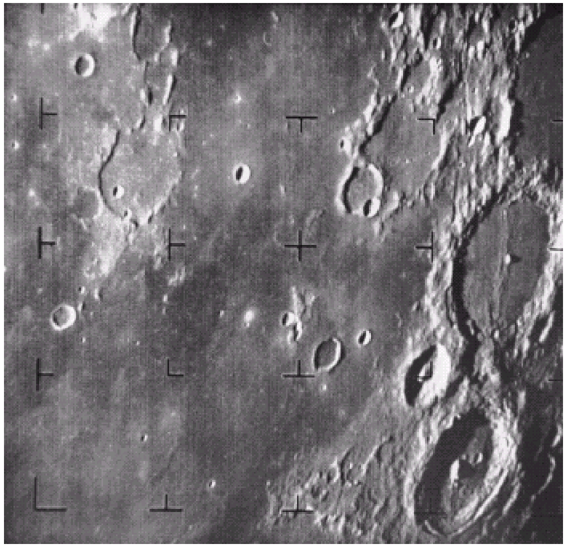


Improved digital image

History of DIP (cont…)

❖ **1960s:** Improvements in **Computing Technology** and the onset of the space race led to a surge of work in Digital Image Processing

- **1964: Computers were** used to improve the quality of images of the moon taken by the *Ranger 7* probe
- Such techniques were used in other space missions including the **Apollo Landings**



A picture of the moon taken by the Ranger 7 probe minutes before landing

History of DIP (cont…)

❖**1970s: Digital Image Processing** begins to be used in medical applications.

- ➢**1979: Sir Godfrey N. Hounsfield & Prof. Allan M. Cormack** share the **Nobel Prize** in medicine for the invention of tomography, the technology behind **Computerised Axial Tomography (CAT)** scans.

- ➢They can show the **soft tissues, blood vessels, and bones** in various parts of the body

Typical head slice CAT image

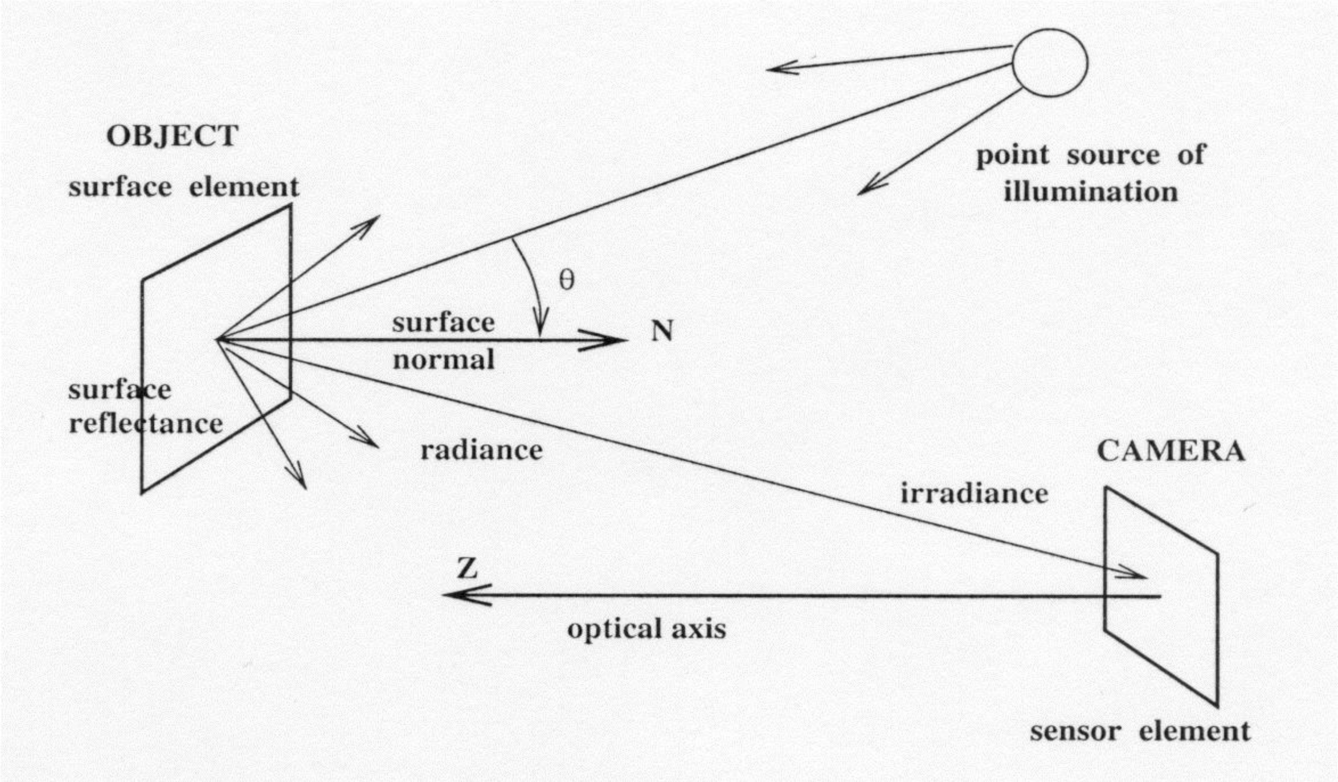
History of DIP (cont…)

❖**1980s - Today:** The use of digital image processing techniques has exploded, and they are now used for all kinds of tasks in all kinds of areas

- Image enhancement/restoration
- Artistic effects
- Medical visualisation
- Industrial inspection
- Law enforcement
- Human computer interfaces
- Data Science
- IoT etc.

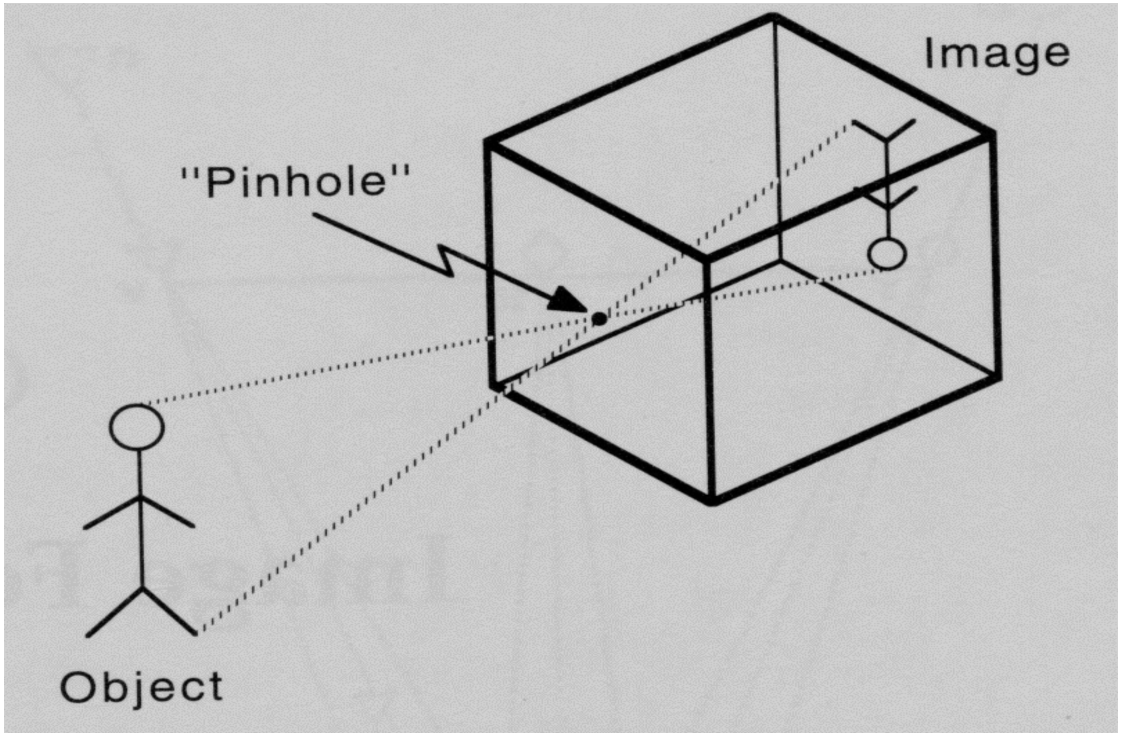
A Simple model of Image formation

- The scene is illuminated by a single source.
- The scene reflects radiation towards the camera.
- The camera senses it via chemicals on film.



Pinhole Camera

- This is the simplest device to form an image of a 3D scene on a 2D surface.
- Straight rays of light pass through a “pinhole” and form an inverted image of the object on the image plane.



Pinhole Camera

❖ Why is the image inverted in a pinhole camera

- Light does not usually bend but travels in a straight line so the light from the top of a scene passes through the pinhole, continues in a straight line, and ends up at the bottom.

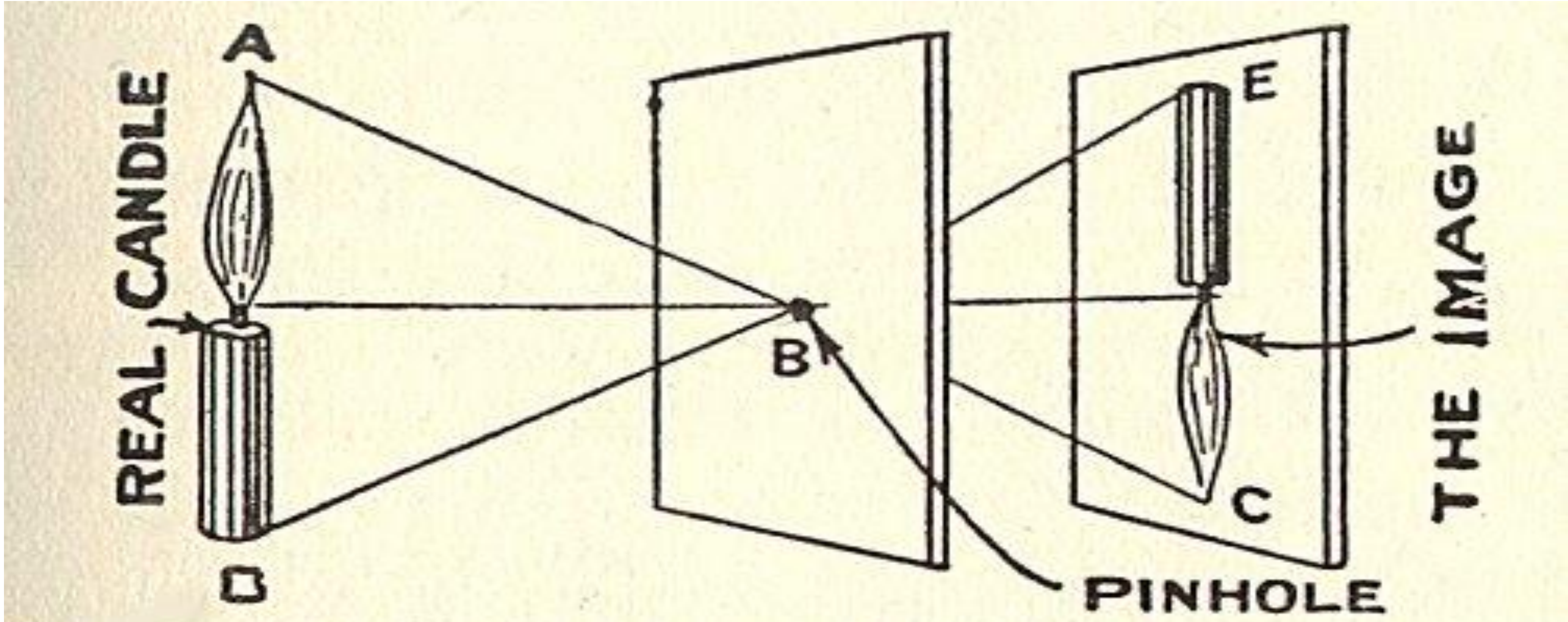


FIG. 131.—How Light and a Pinhole Form an Image.

Camera Optics

- In practice, the aperture must be larger to admit more light.
- Lenses are placed to in the aperture to **focus** the bundle of rays from each scene point onto the corresponding point in the image plane

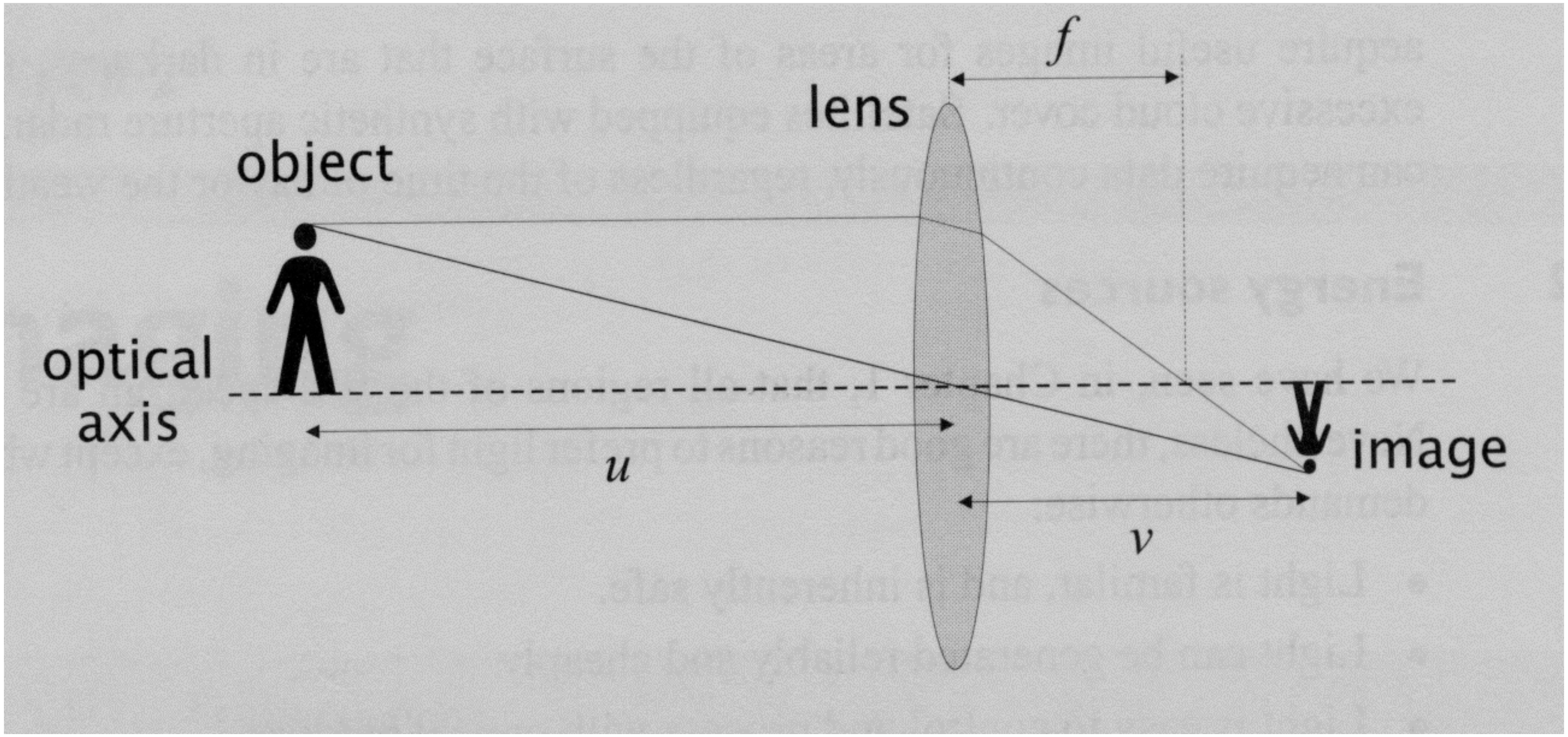


Image Formation

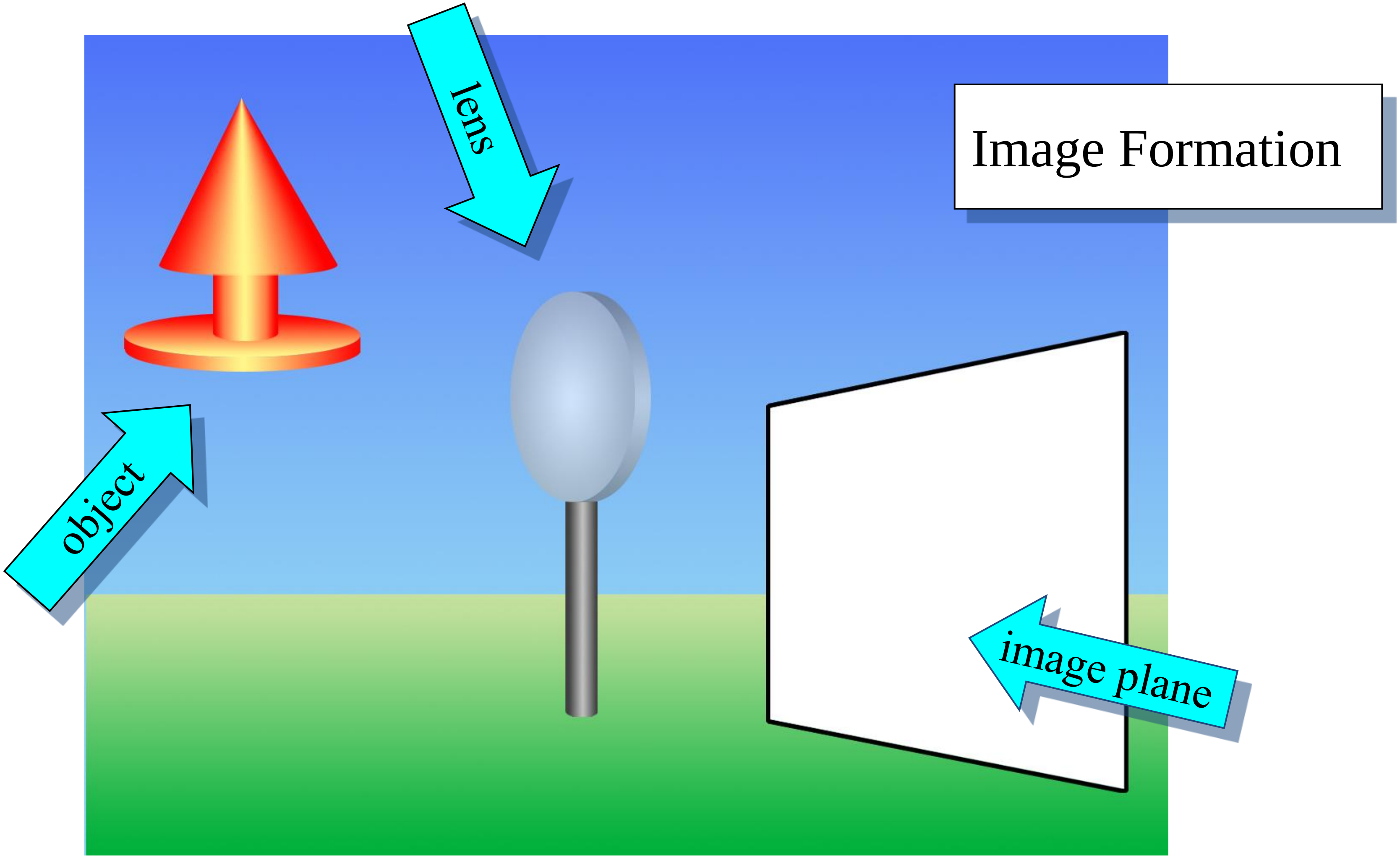


Image Formation

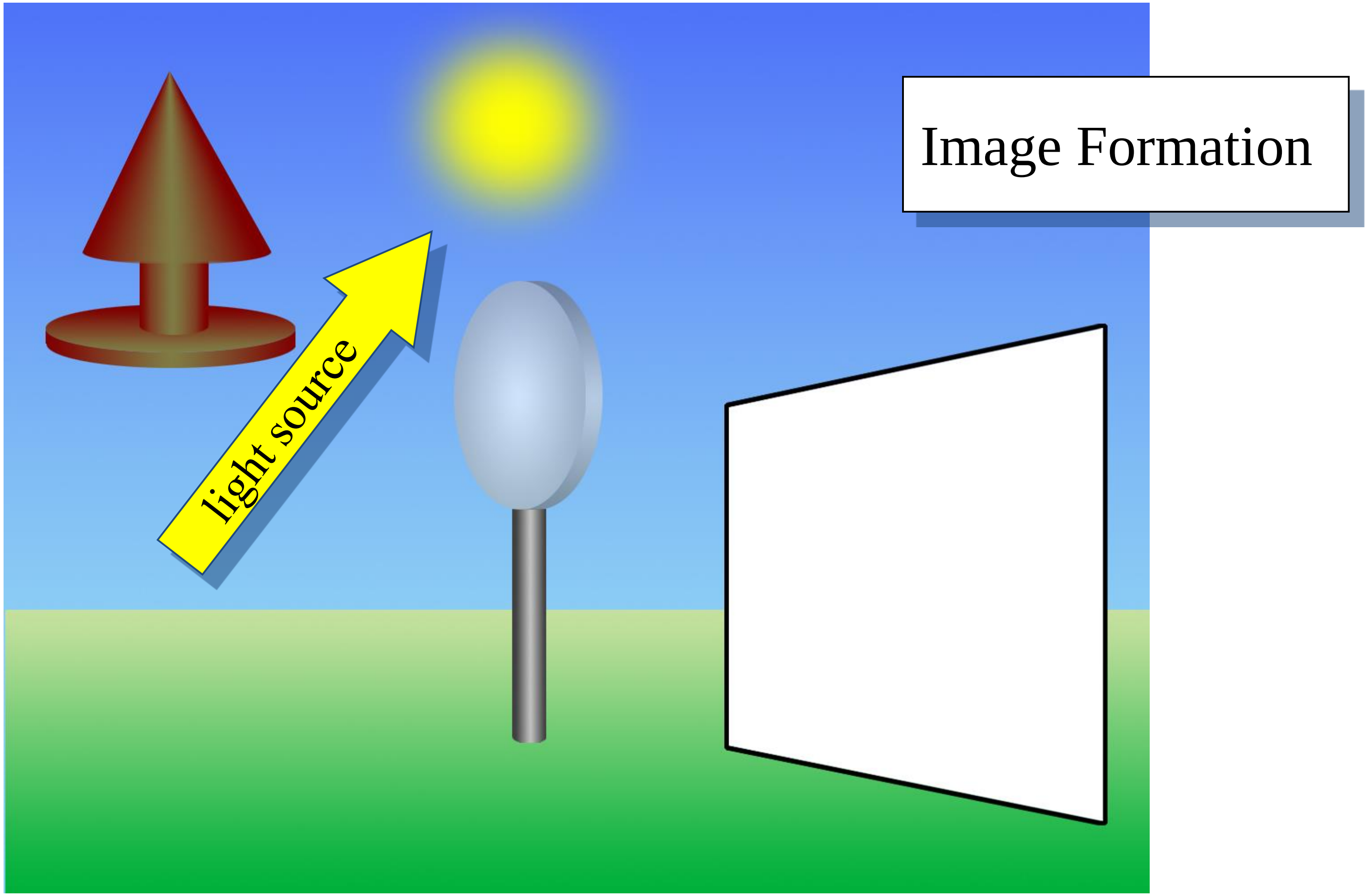


Image Formation

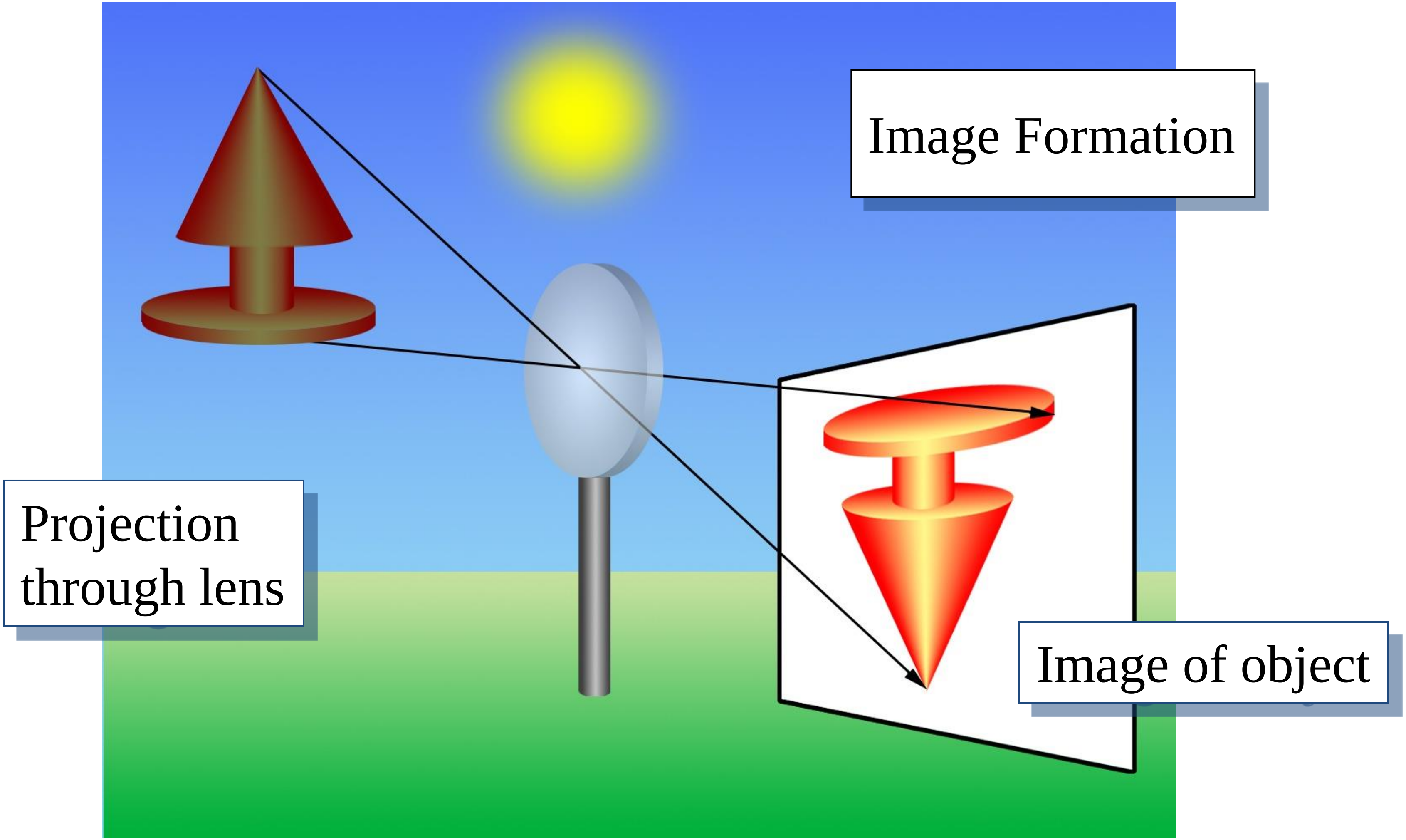


Image Formation

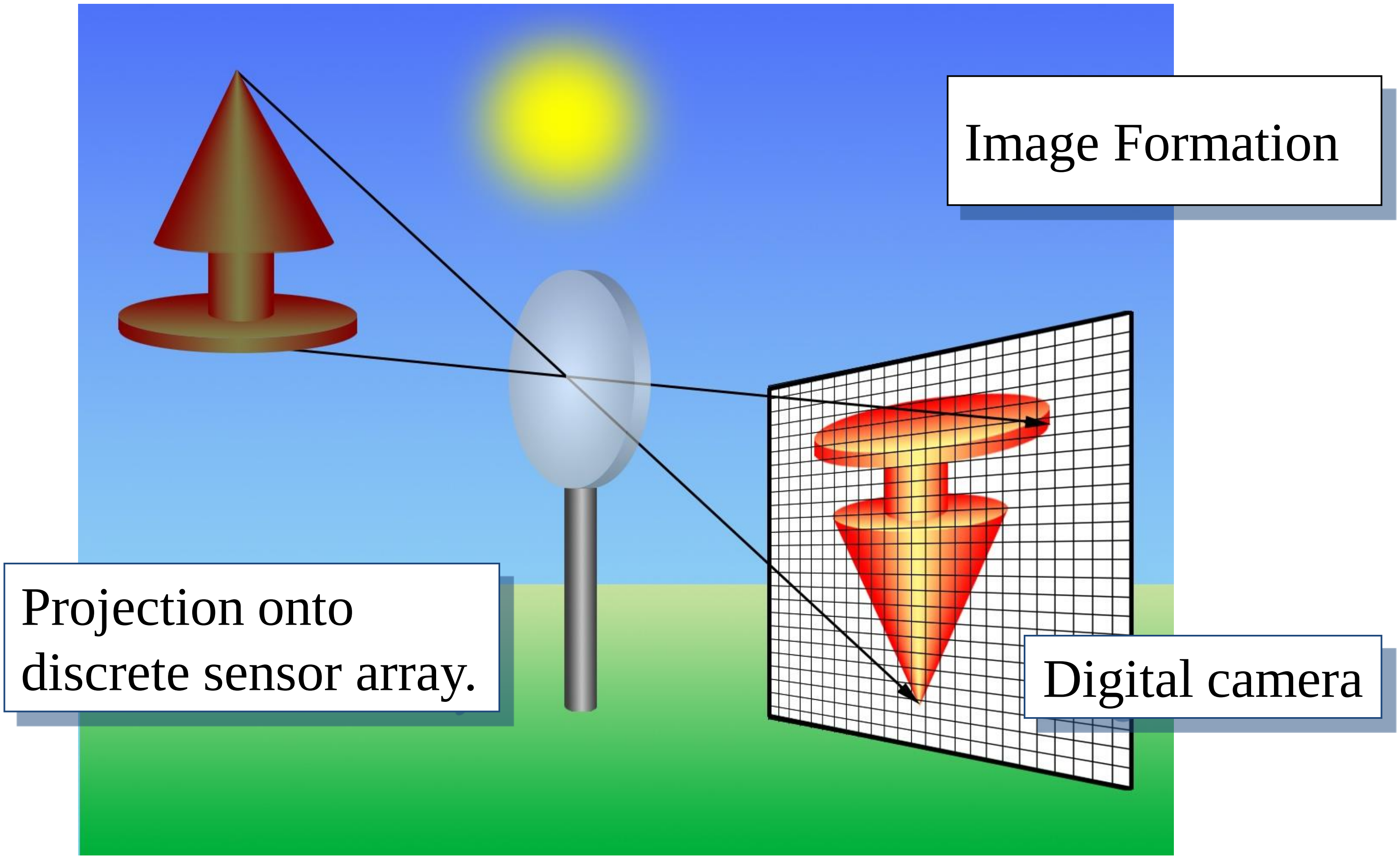


Image Formation

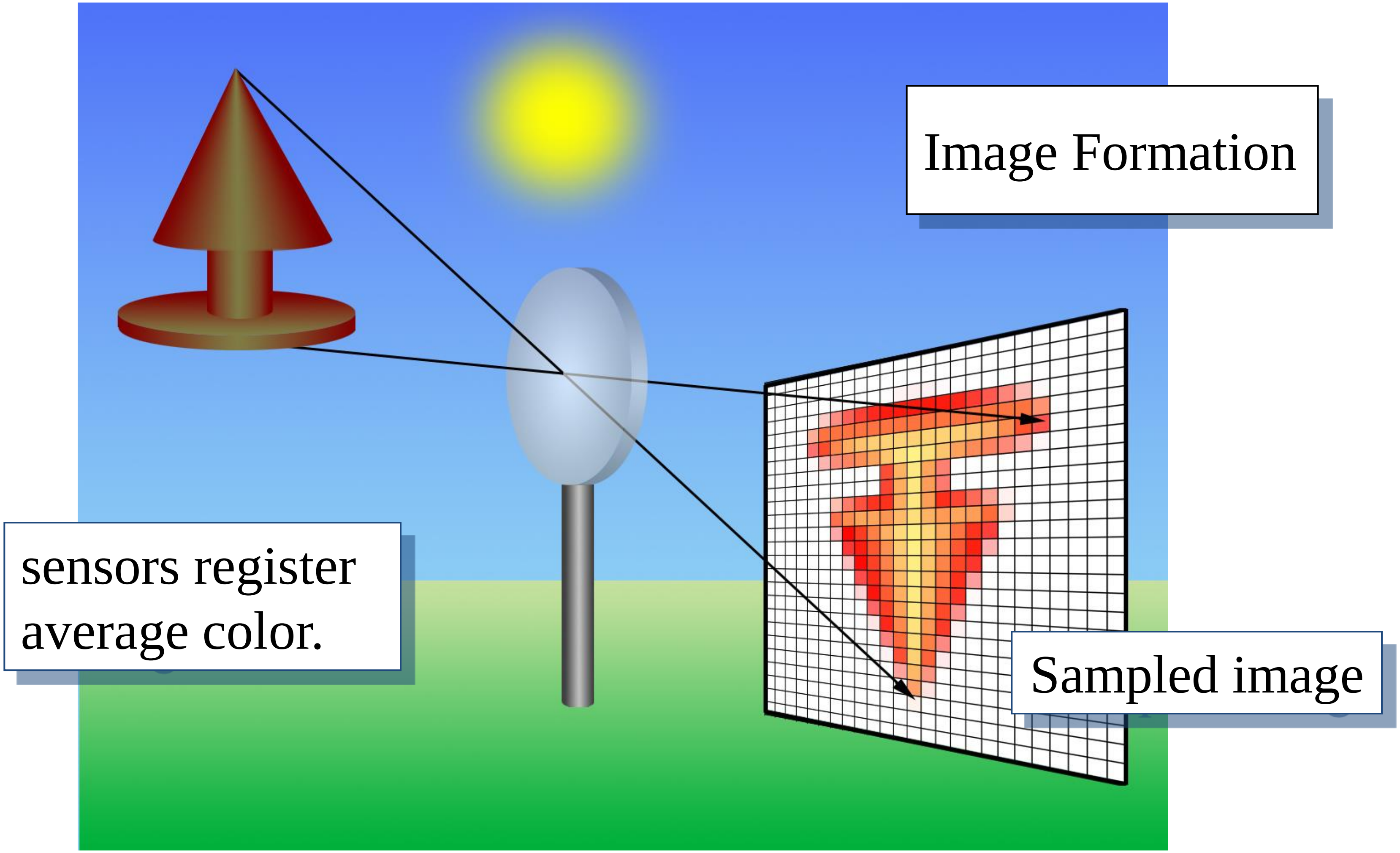
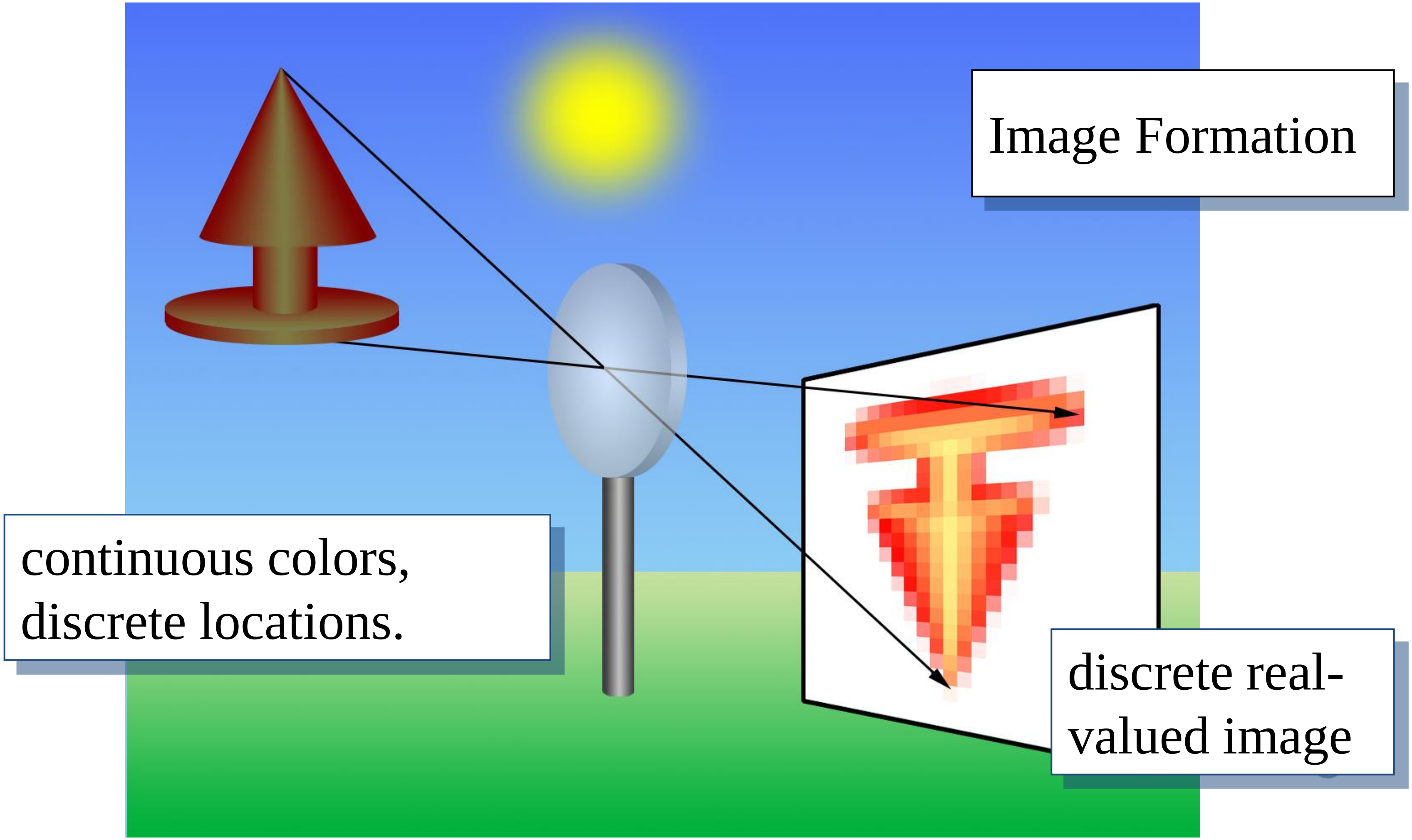
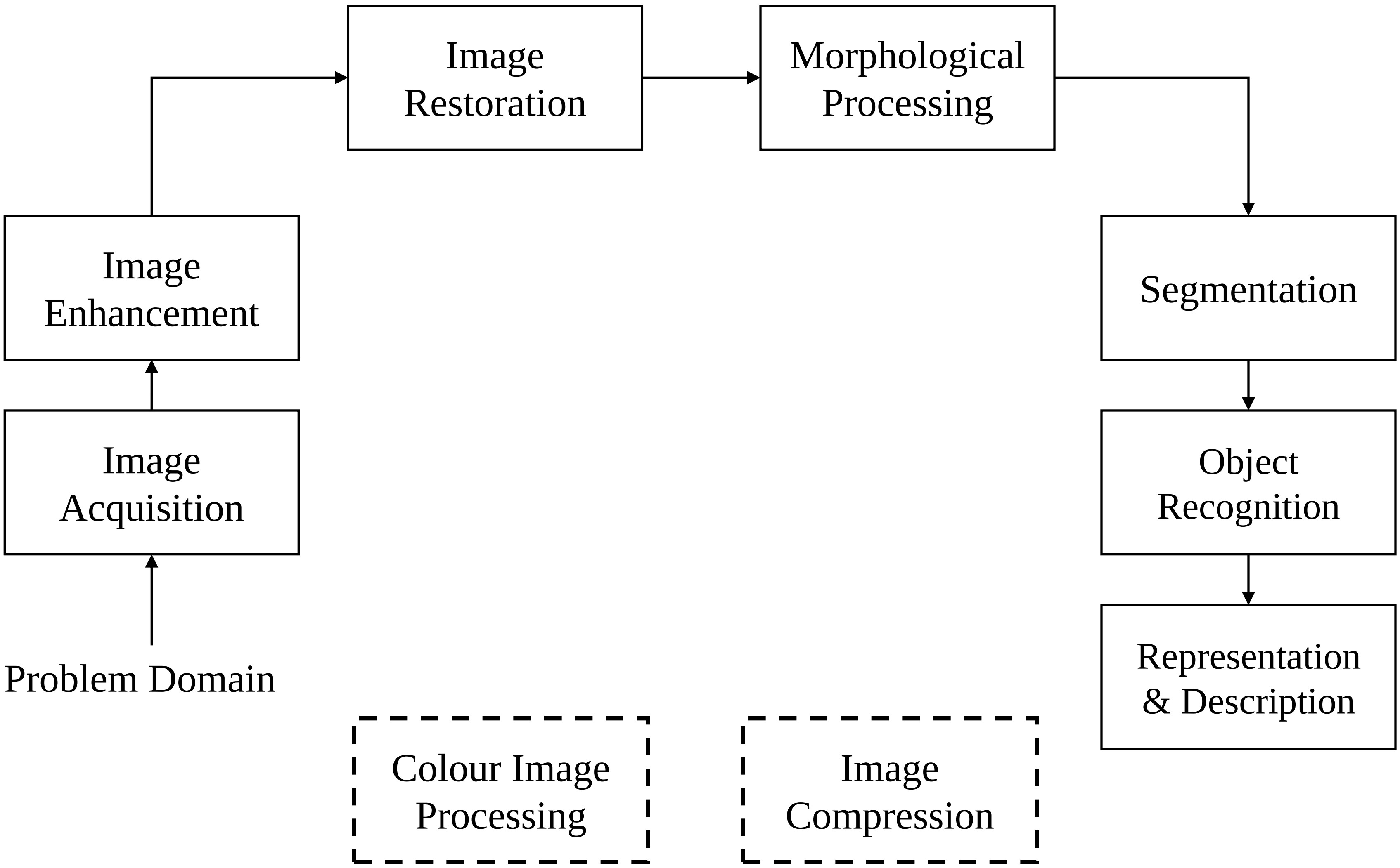


Image Formation



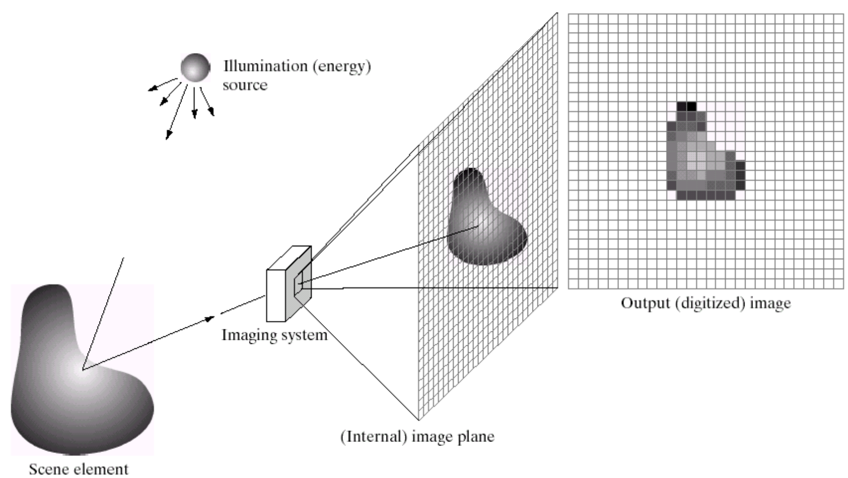
Key Stages in Digital Image Processing



Key Stages: Image Acquisition

Image Restoration → Morphological Processing → Segmentation → Object Recognition → Representation & Description

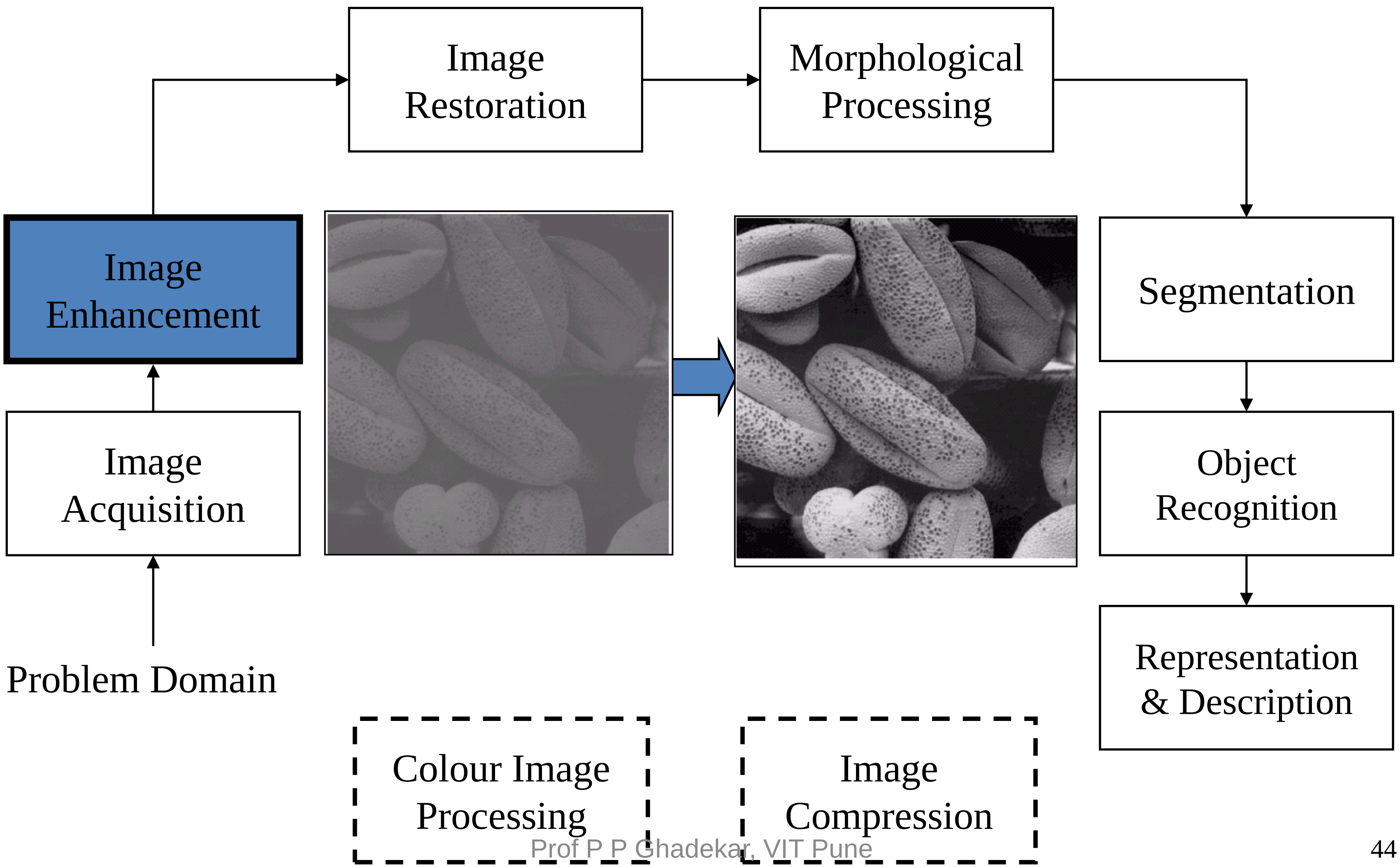
Problem Domain → **Image Acquisition** → Image Enhancement → Image Restoration



Colour Image Processing

Image Compression

Key Stages :Image Enhancement

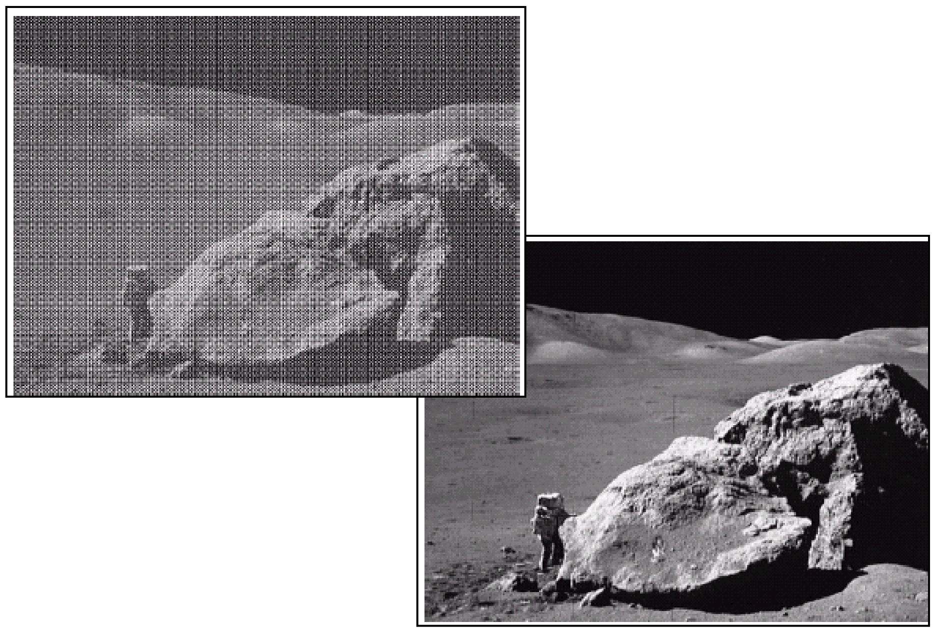


Key Stages :Image Restoration

Image Restoration

Morphological Processing

Image Enhancement



Segmentation

Image Acquisition

Object Recognition

Representation & Description

Problem Domain

Colour Image Processing

Image Compression

Key Stages :Morphological Processing

Image Restoration

Morphological Processing

Image Enhancement

Image Acquisition

Problem Domain

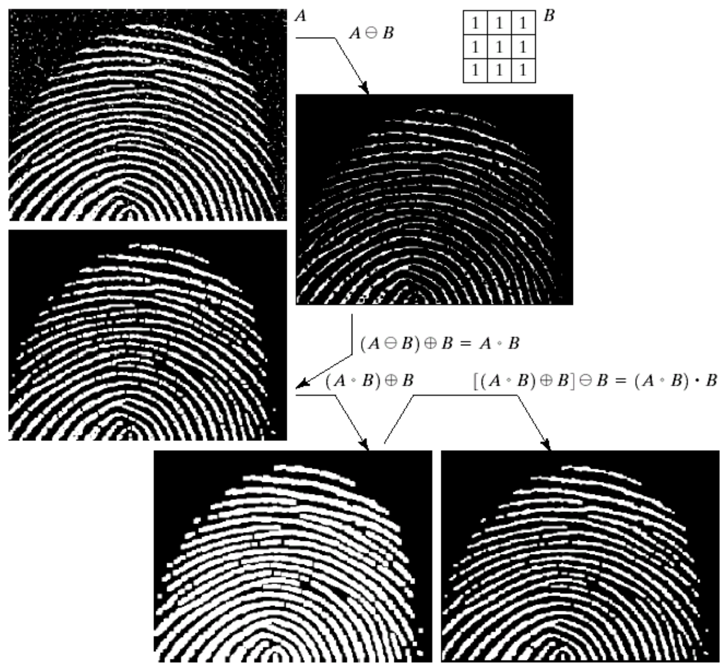
Segmentation

Object Recognition

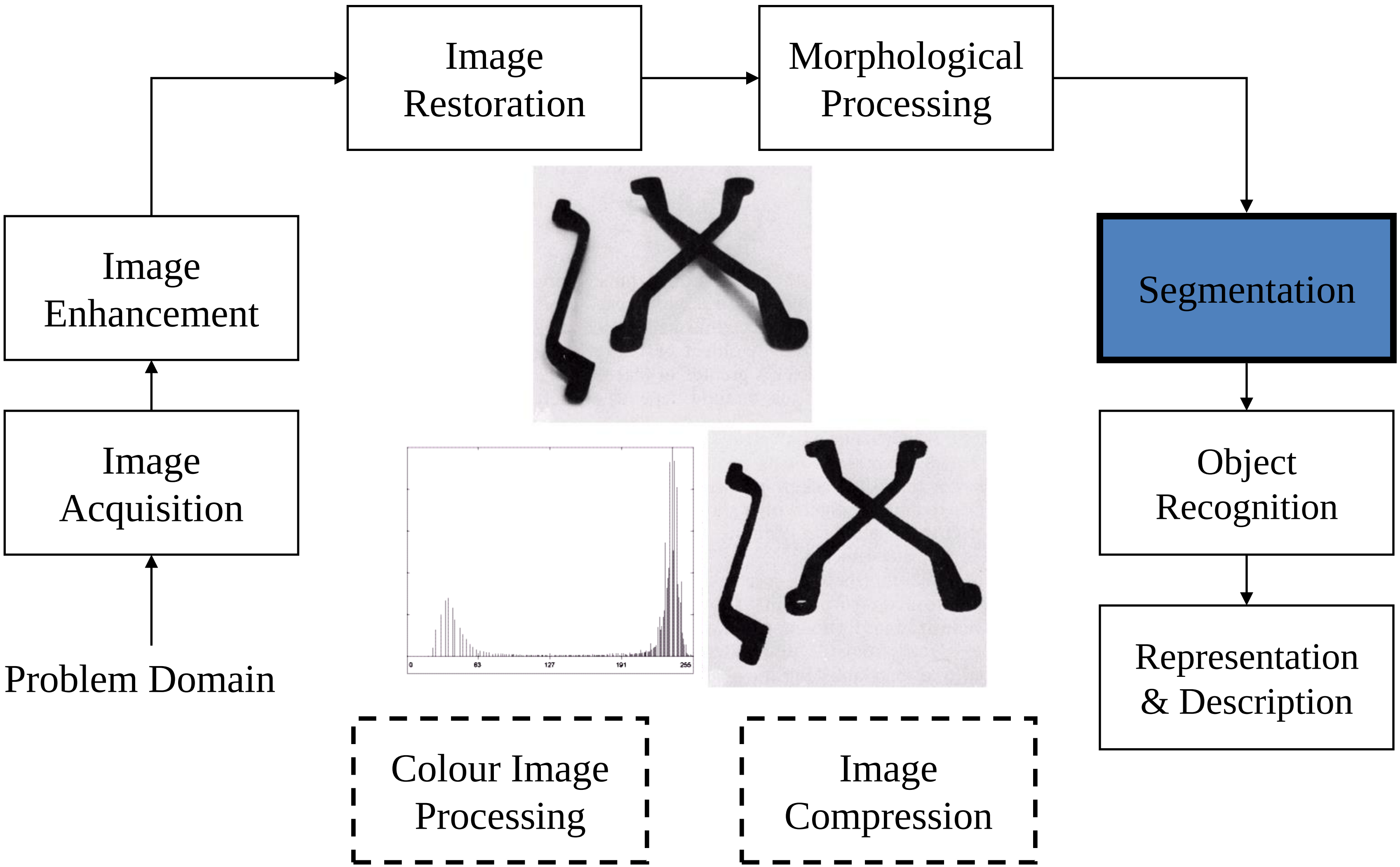
Representation & Description

Colour Image Processing

Image Compression



Key Stages :Segmentation



Key Stages :Object Recognition

Image Restoration

Morphological Processing

Image Enhancement

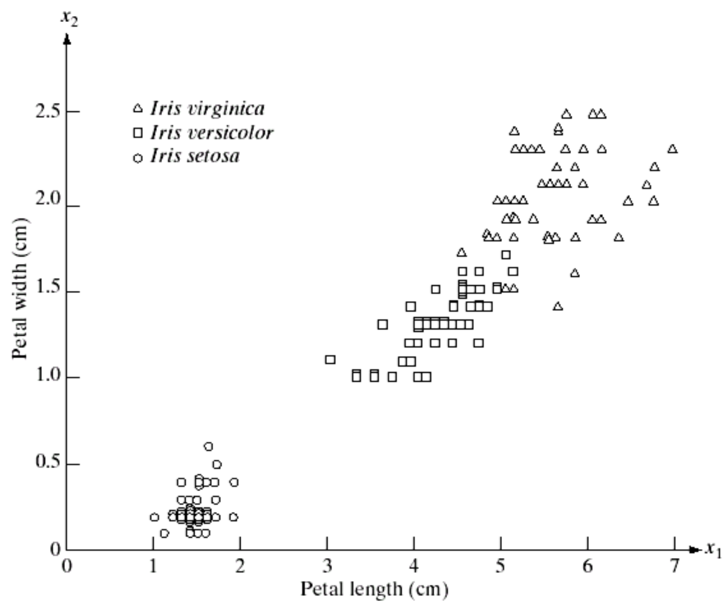
Segmentation

Image Acquisition

Object Recognition

Problem Domain

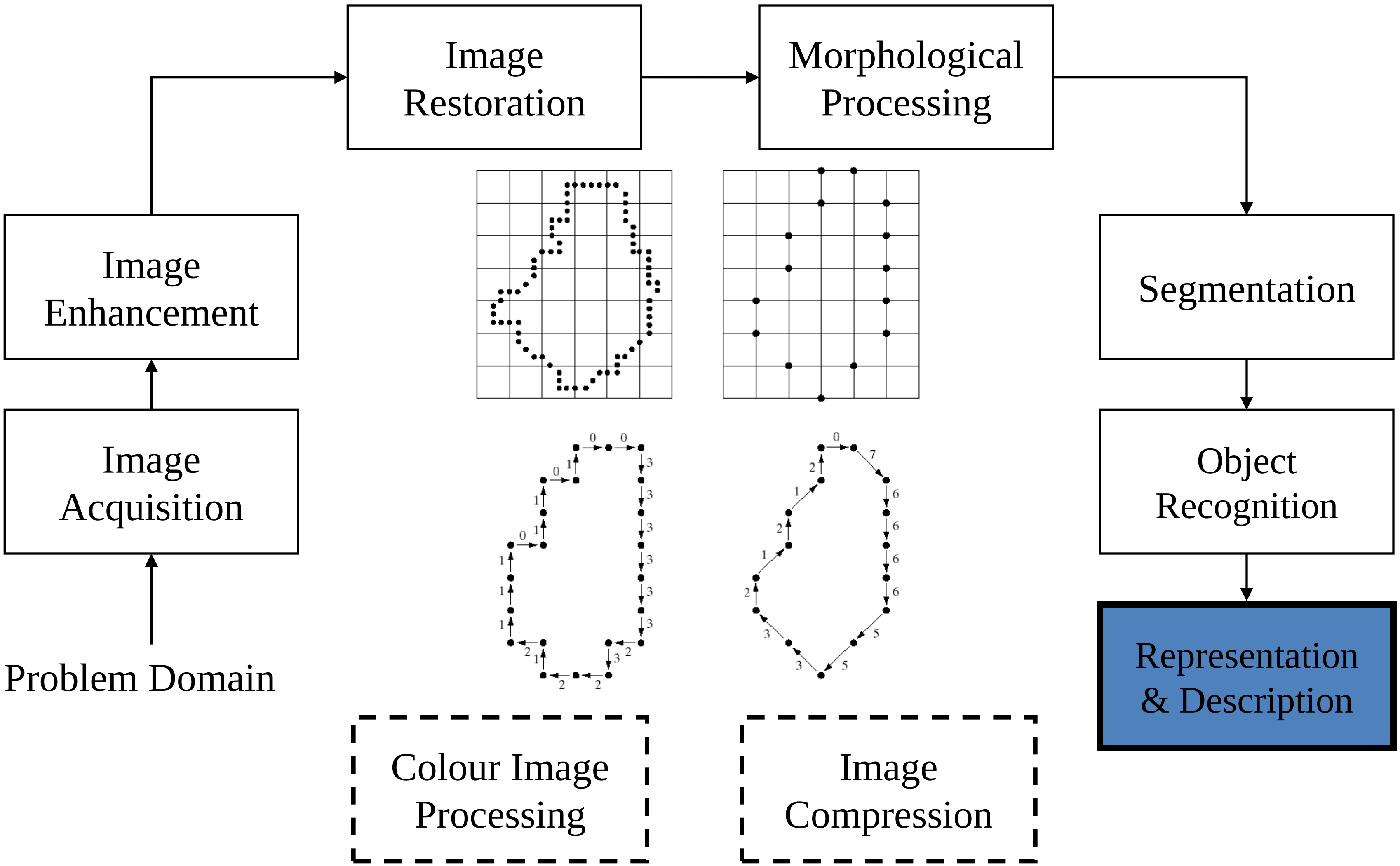
Representation & Description



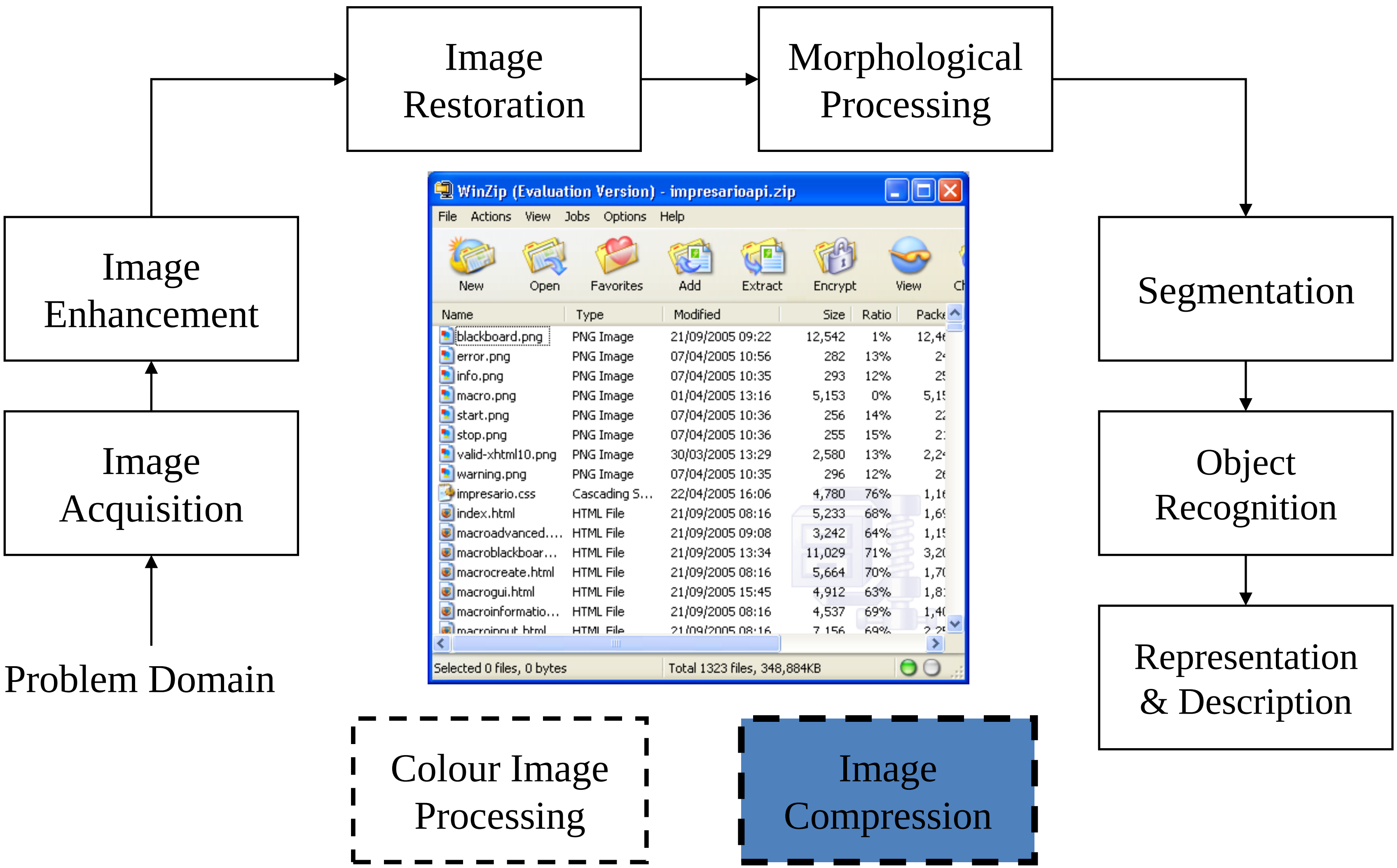
Colour Image Processing

Image Compression

Key Stages :Representation & Description



Key Stages :Image Compression

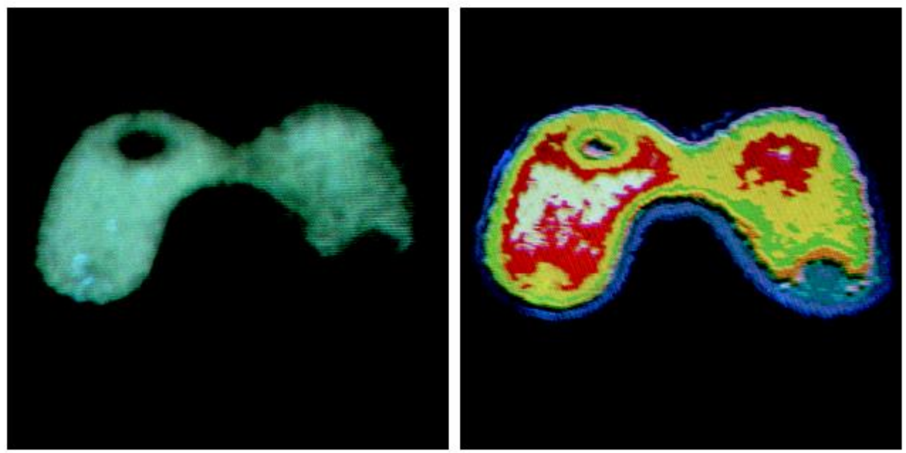


Key Stages :Colour Image Processing

Image Restoration

Morphological Processing

Image Enhancement



Segmentation

Image Acquisition

Object Recognition

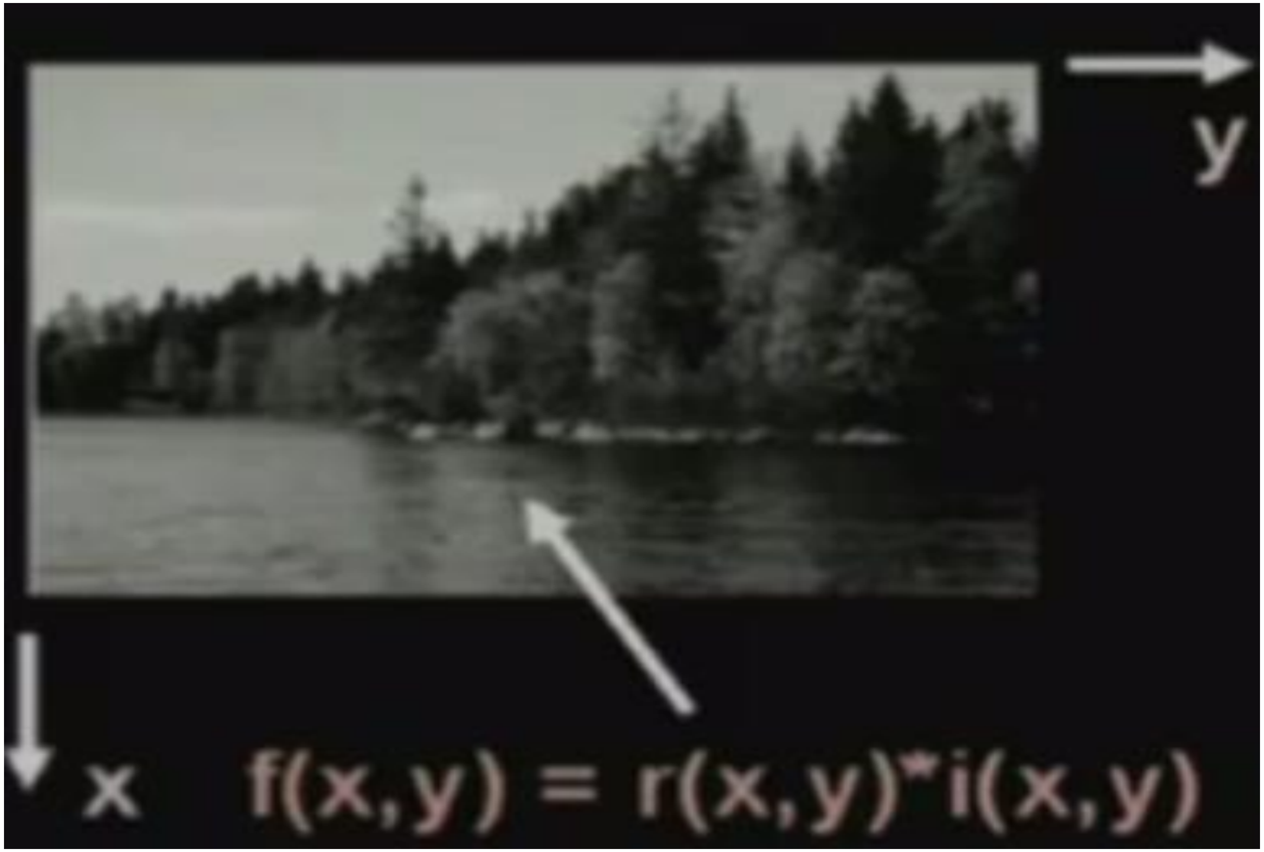
Problem Domain

Representation & Description

Colour Image Processing

Image Compression

Image Representation



$$F(x,y)=r(x,y)*i(x,y)$$

Where r(x,y)-reflectivity of surface and i(x,y) –amount of intensity of the light incident on the object.

Image Representation

$$I=\begin{bmatrix} f(0,0) & f(0,1) & \ldots\ldots\ldots\ldots & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots\ldots\ldots\ldots & f(1,N-1) \\ f(2,0) & f(2,1) & \ldots\ldots\ldots\ldots & f(2,N-1) \\ . & & & \\ . & & & \\ . & & & \\ . & & & \\ f(M-1,0) & f(M-1,1) & \ldots\ldots & f(M-1,N-1) \end{bmatrix}$$

Image Processing Applications

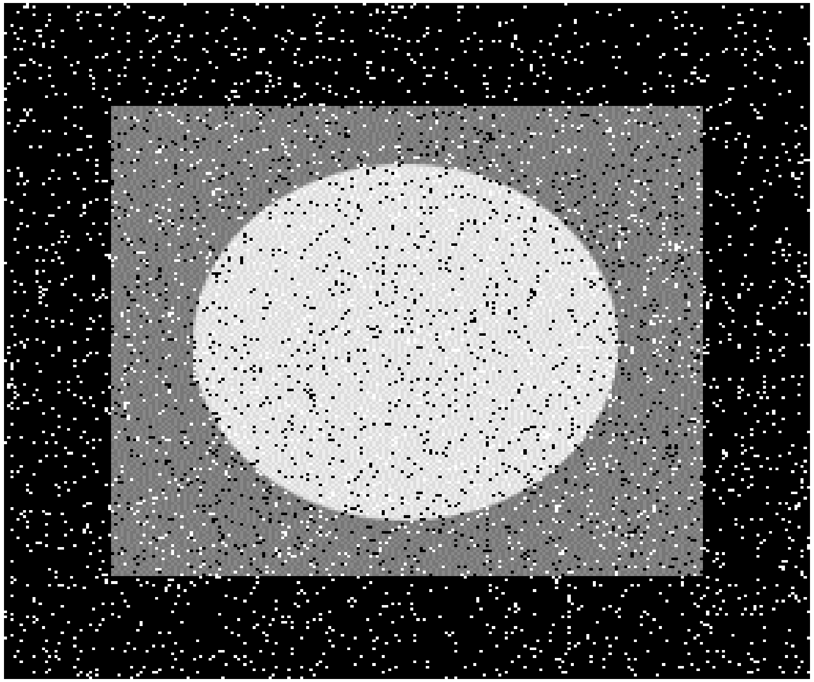
- Human Perception
- Medical Applications
- Weather Forecasting
- Human Computer Interface
- Industrial Applications
- Military Applications
- Law Enforcement and Security
- Consumer Electronics
- The Internet, Particularly the World Wide Web

Human Perception

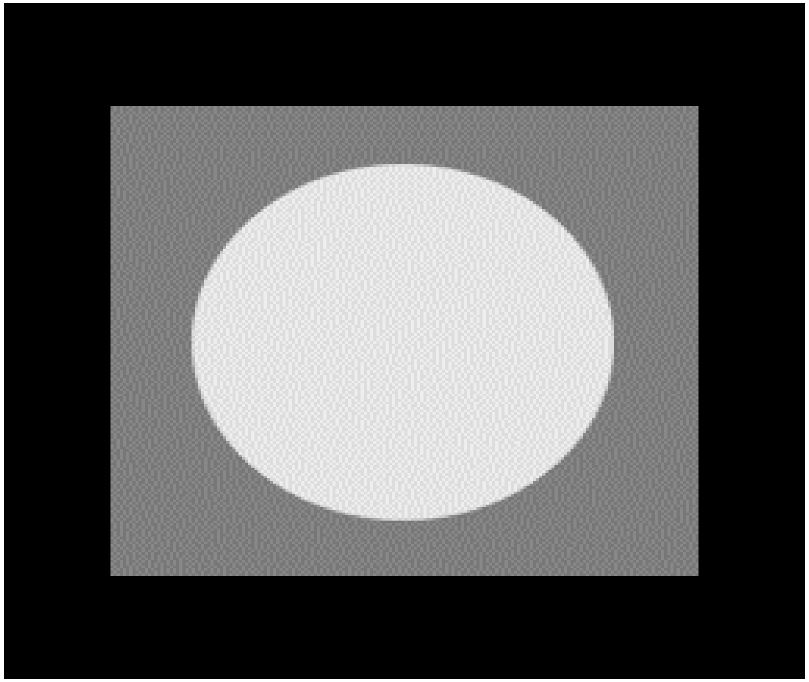
❖ Typical Application

- Noise Filtering
- Content Enhancement
- Contrast Enhancement
- Deblurring
- Remote sensing

Filtering



Noisy Image



Filtered Image

Image Enhancement

Low Contrast Image

Enhanced Image

Image Enhancement

Low Contrast Image

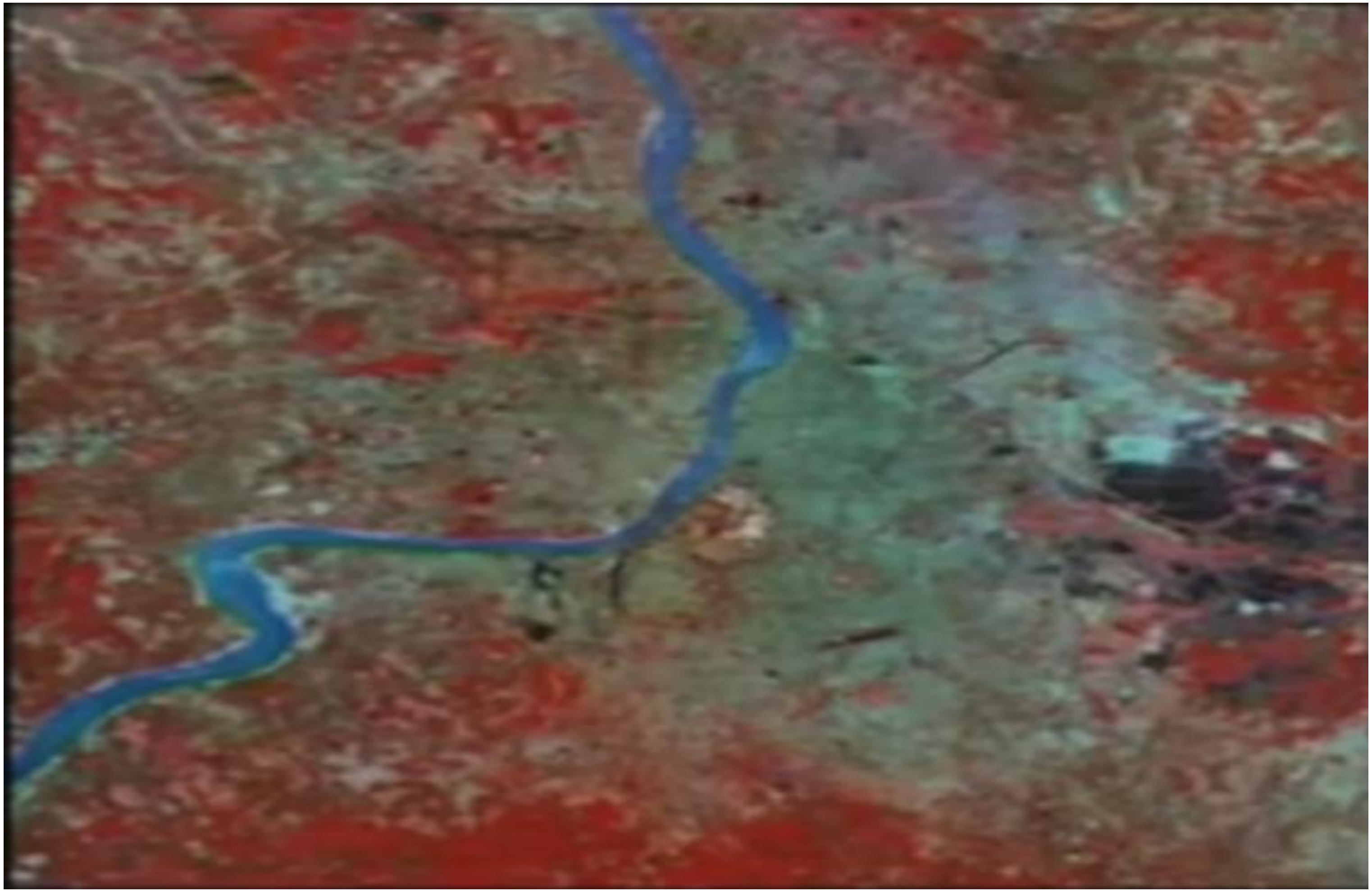
Enhanced Image

Deblurring

Blurred Image

DeBlurred Image

Remote Sensing



Satellite Image Kolkata

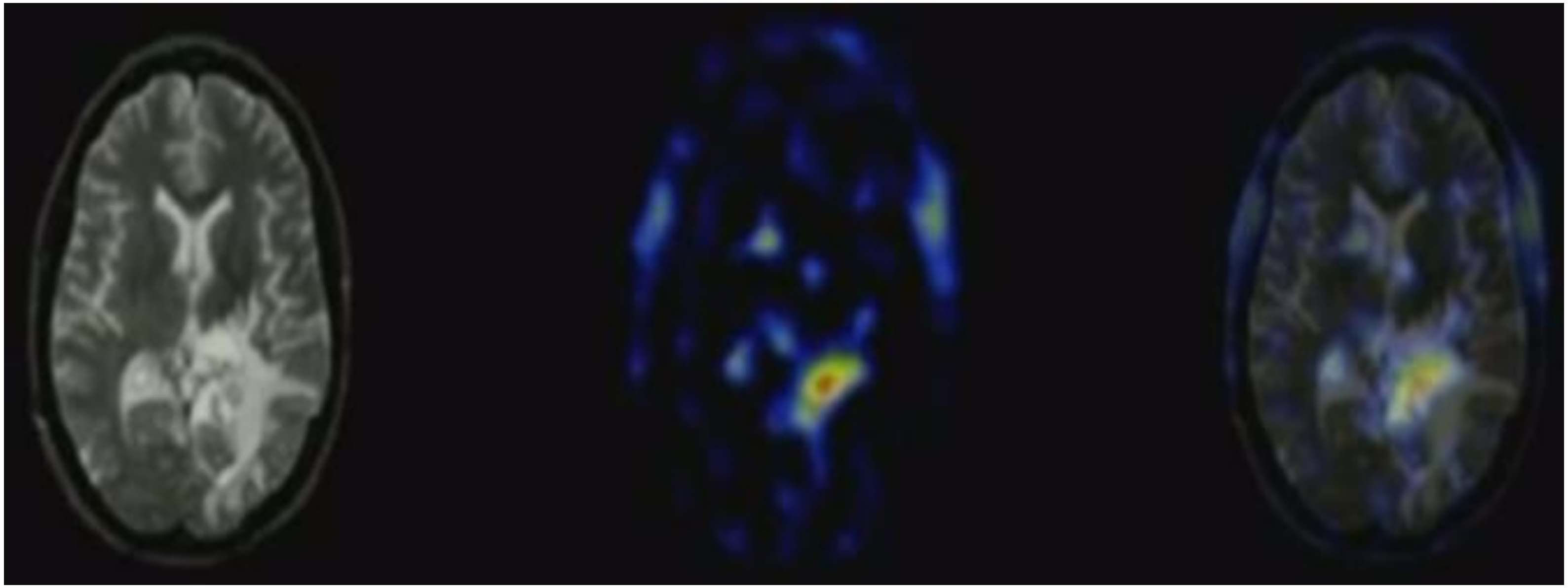
Remote Sensing

Terrain Mapping of Hilly region

Remote Sensing

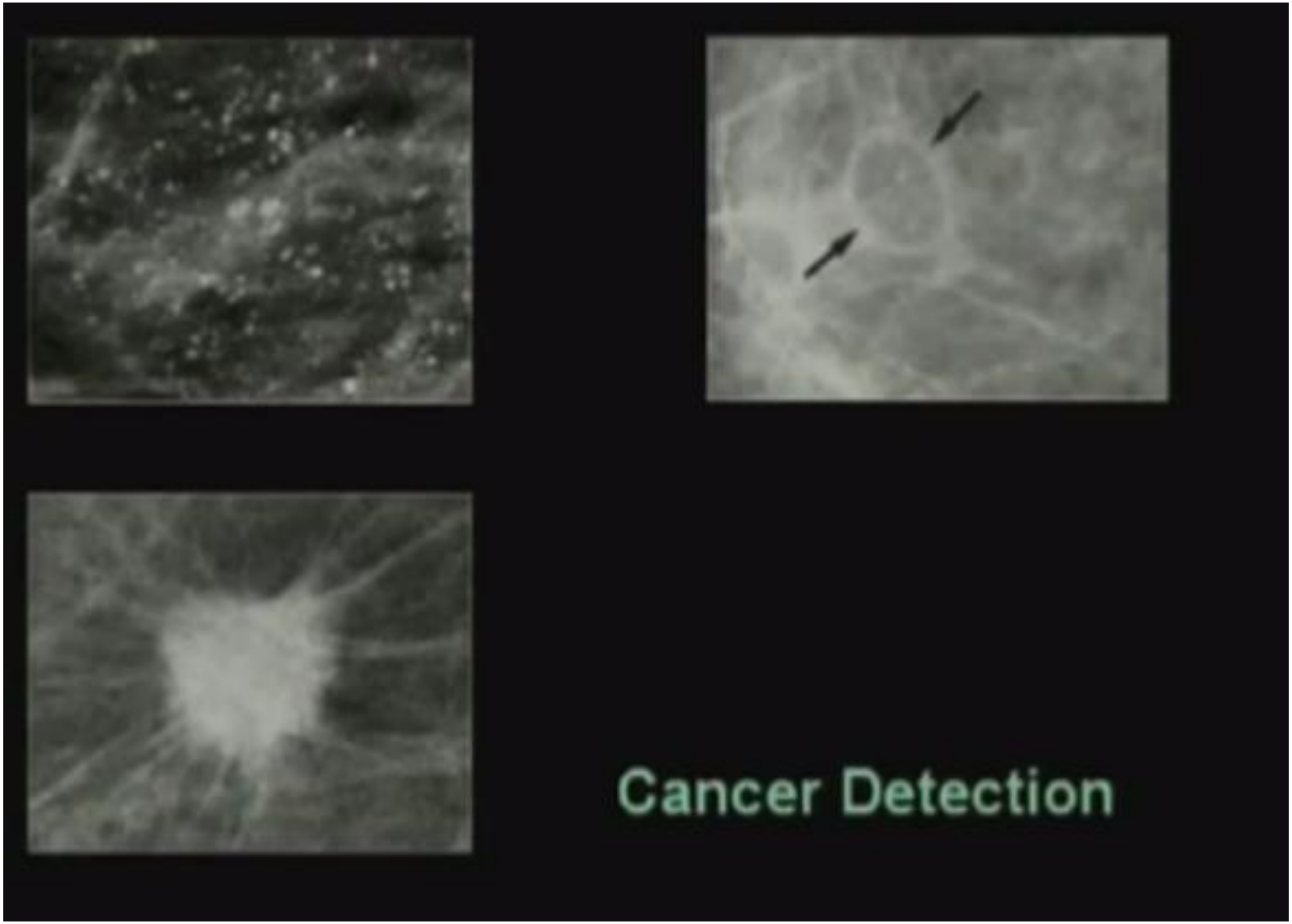
Borneo Fire

Medical Imaging



One Slice of CT scan Image- Brain Tumour

Medical Imaging

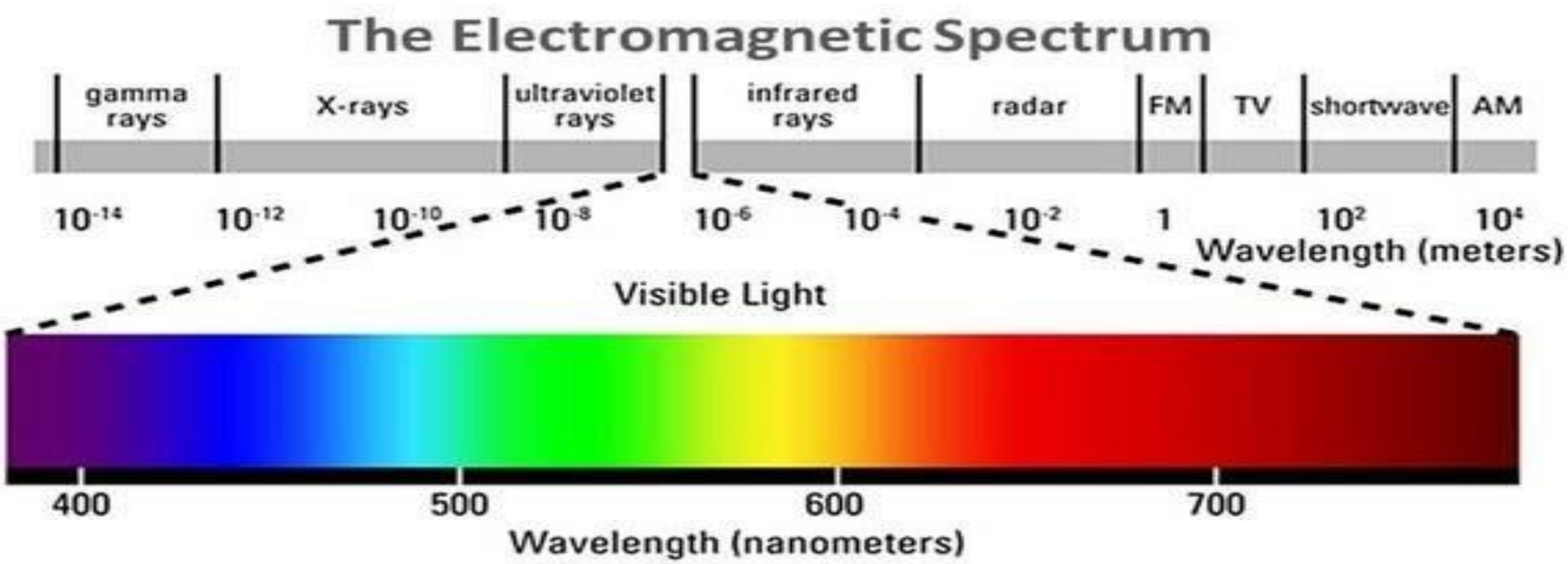


Medical Imaging

Ultra Sonogram

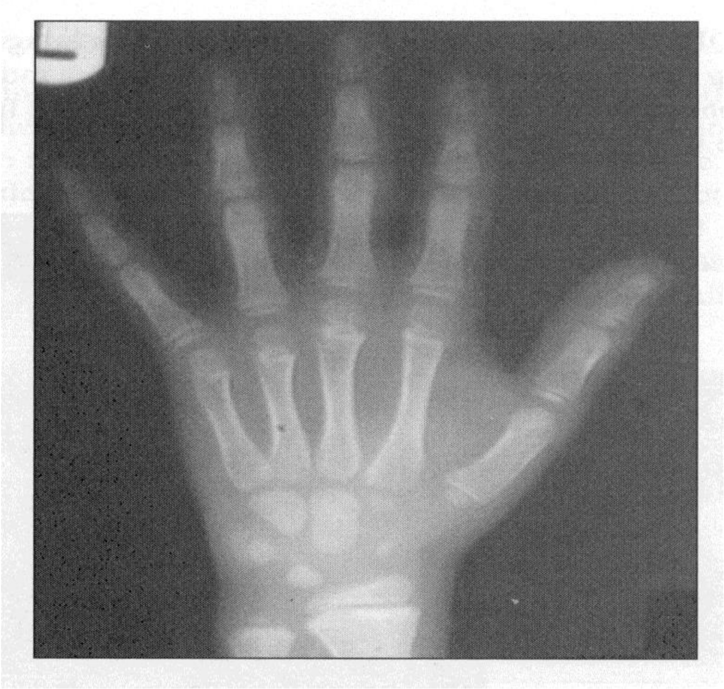
What is light?

- The visible portion of the **electromagnetic** (EM) spectrum.
- It occurs between wavelengths of approximately 400 and 700 nanometers.



Short Wavelengths

- Different wavelengths of radiation have different properties.
- The **x-ray** region of the spectrum, it carries sufficient energy to penetrate a significant volume or material.



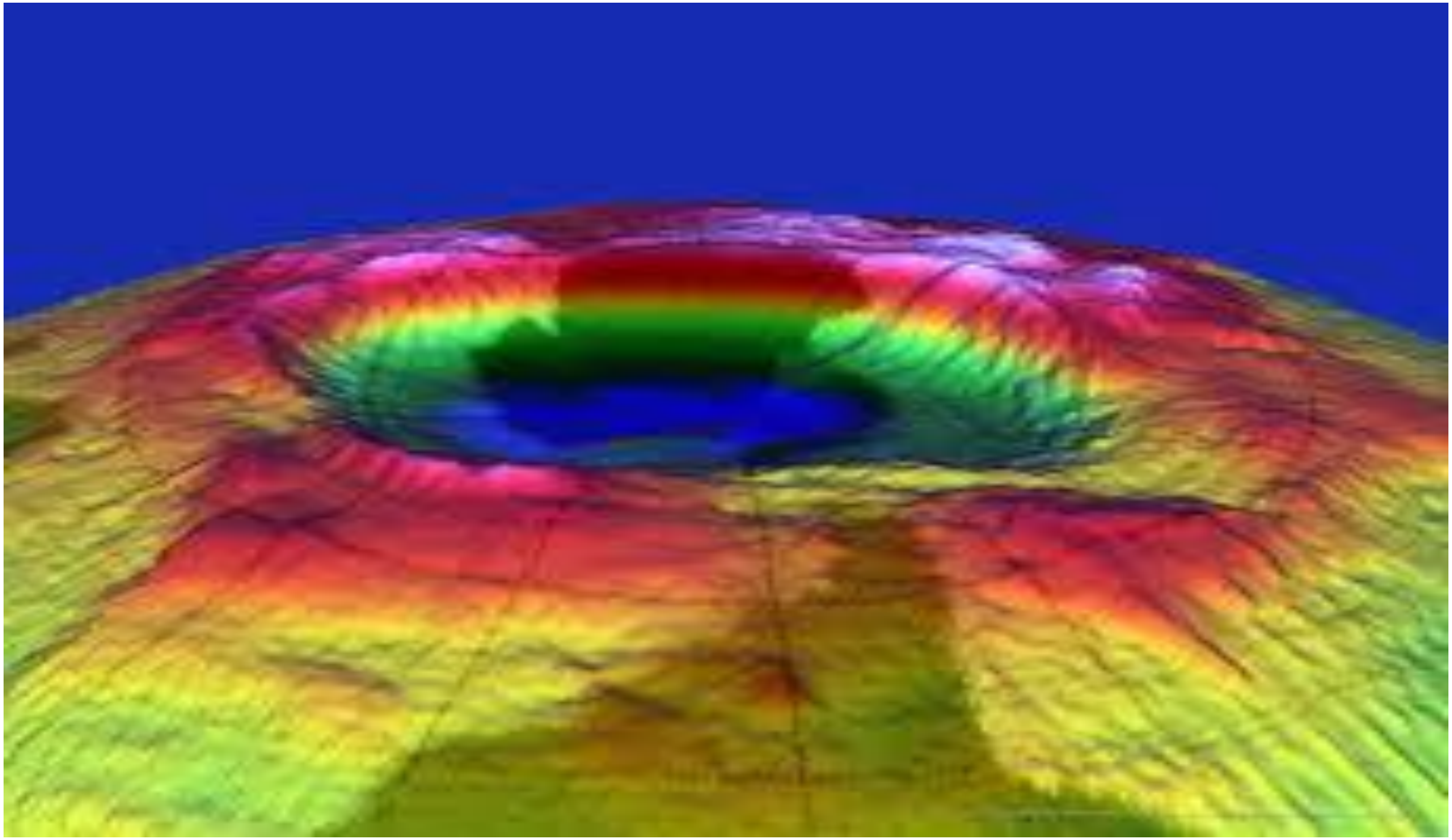
Long Wavelengths

- Many quantities of **infrared** (IR) radiation are emitted from warm objects (e.g., locate people in total darkness).

Long Wavelengths (cont’d)

- “**<u>Synthetic aperture radar</u>**” (SAR) imaging techniques use an artificially generated source of microwaves to probe a scene.
- SAR is unaffected by weather conditions and clouds (e.g., has provided us images of the surface of Venus).

Atmospheric Study



Ozone Hole

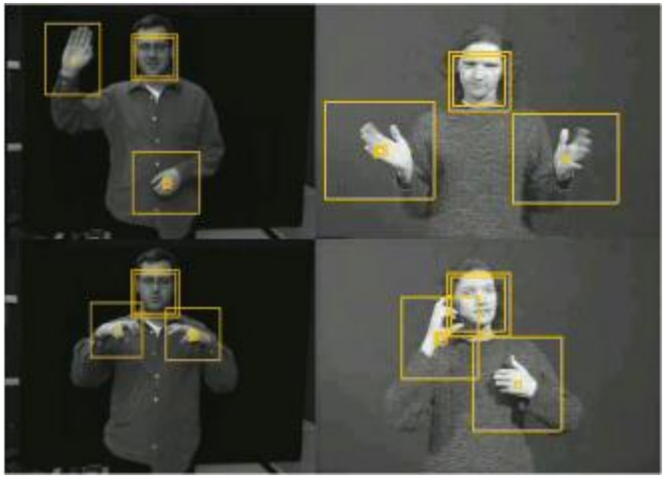
Human Computer Interface

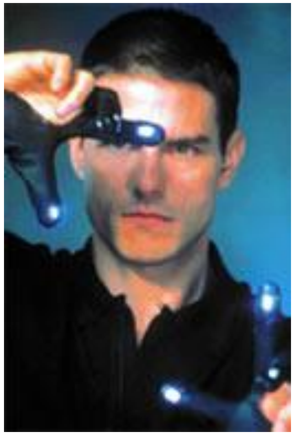
❖Try to make human computer interfaces more natural

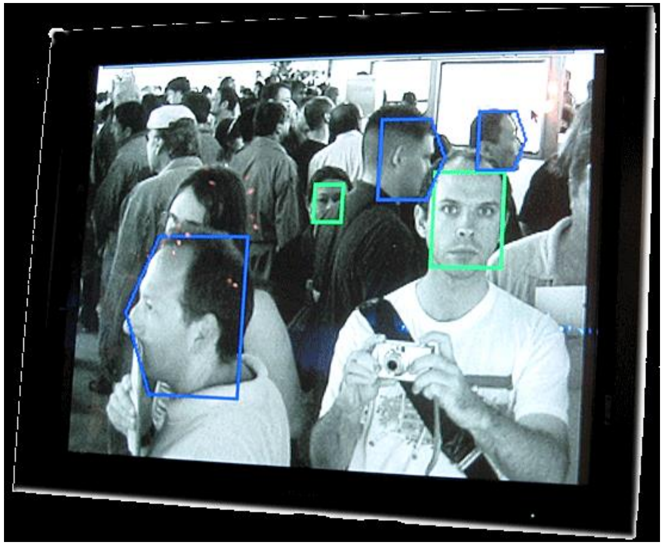
- ➢ Face recognition
- ➢ Gesture recognition

❖Does anyone remember the user interface from “Minority Report”?

•These tasks can be extremely difficult







Examples: Artistic Effects

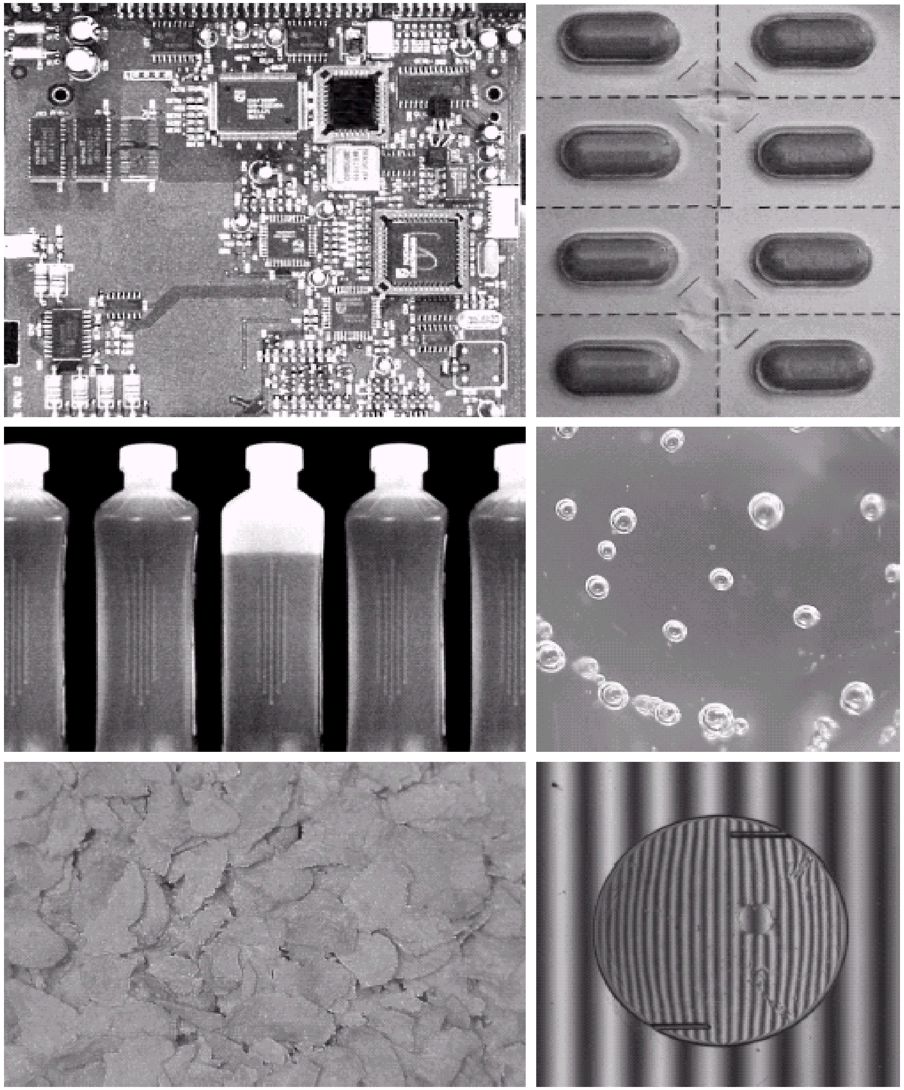
❖Artistic effects are used to make images more visually appealing, to add special effects and to make composite images.

Examples: Industrial Inspection

❖Human operators are expensive, slow and unreliable

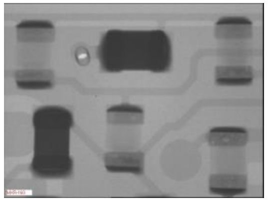
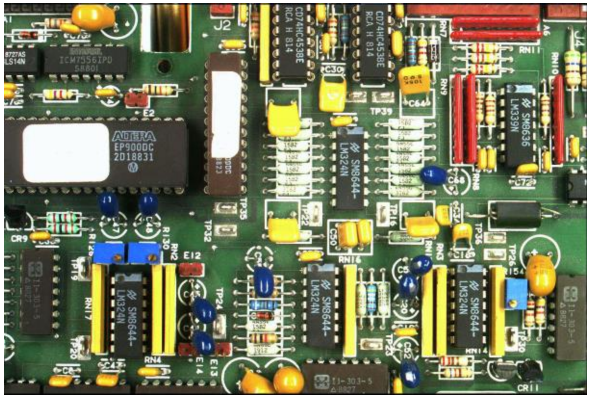
❖Make machines do the job instead

❖Industrial vision systems are used in all kinds of industries



Examples: PCB Inspection

➢ Machine inspection is used to determine that all components are present and that all solder joints are acceptable

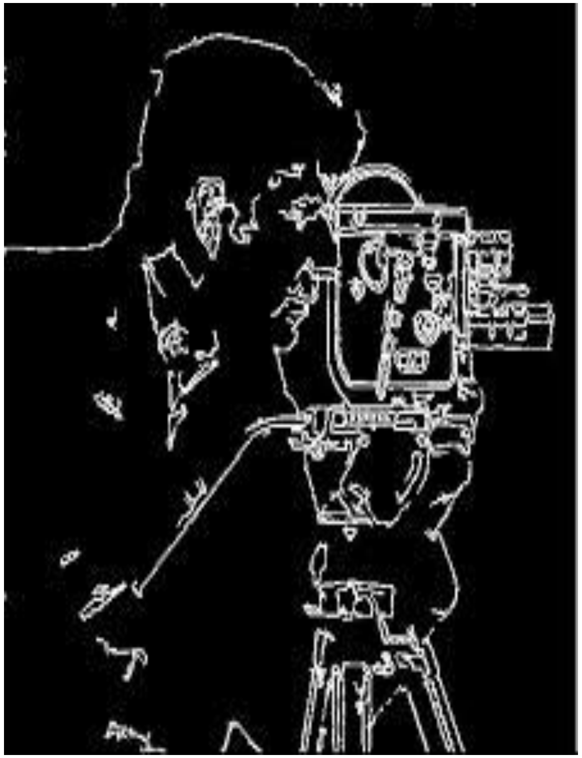

The Internet, Particularly the World Wide Web

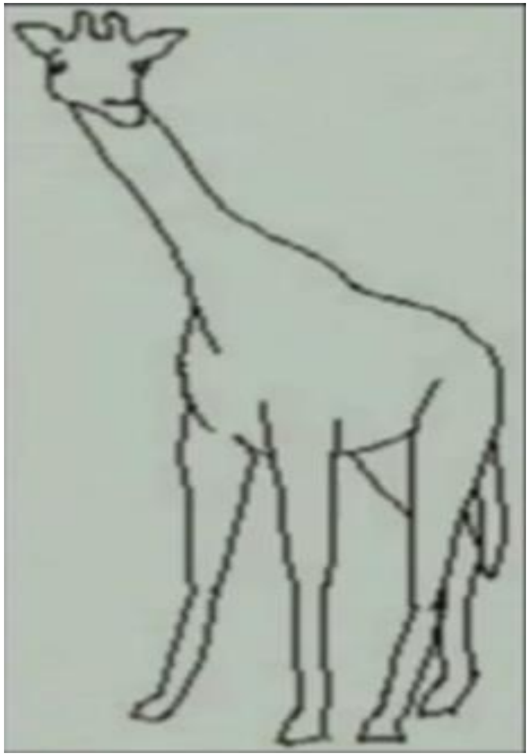
- There is a huge amount of visual information available on the Web.
- Collaborative image and video uploading, sharing, and annotation (tagging) have become increasingly popular.
- Finding and retrieving images and videos on the Web based on their contents.

Astronomy

Galaxy

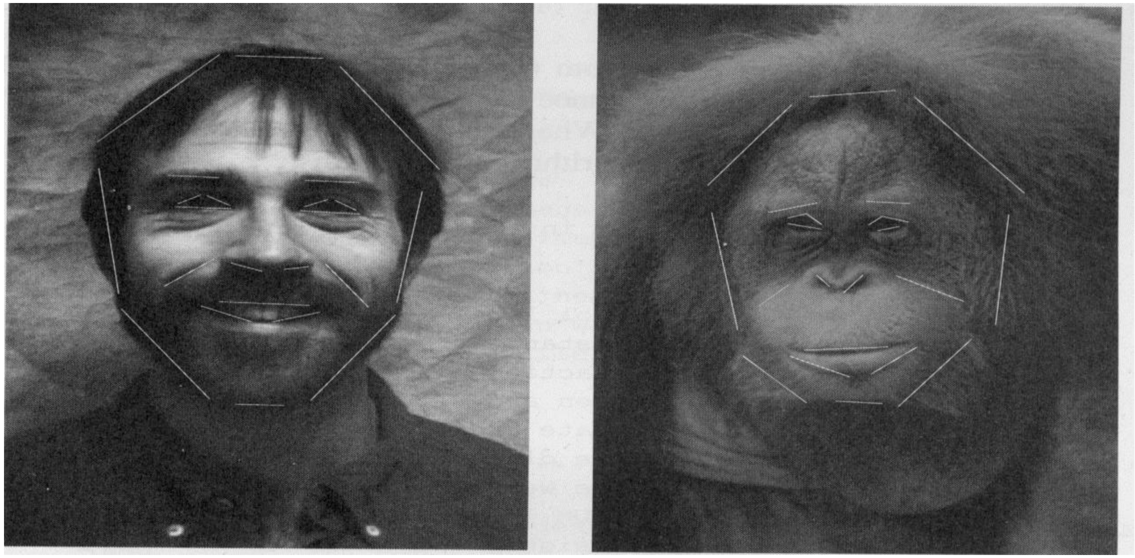
Boundary Information





Importance of Boundary Information

Morphing



Inserting Artificial Objects into a Scene

Image Addition

+

=

Image-1

Image-2

Output Image

Background Subtraction

Image-1

Image-2

Absolute Difference

Image and Video Compression

Image and video data compression refers to a process in which the amount of data used to represent image and video is reduced to meet a bit rate requirement.

Functionality in Visual Transmission and Storage

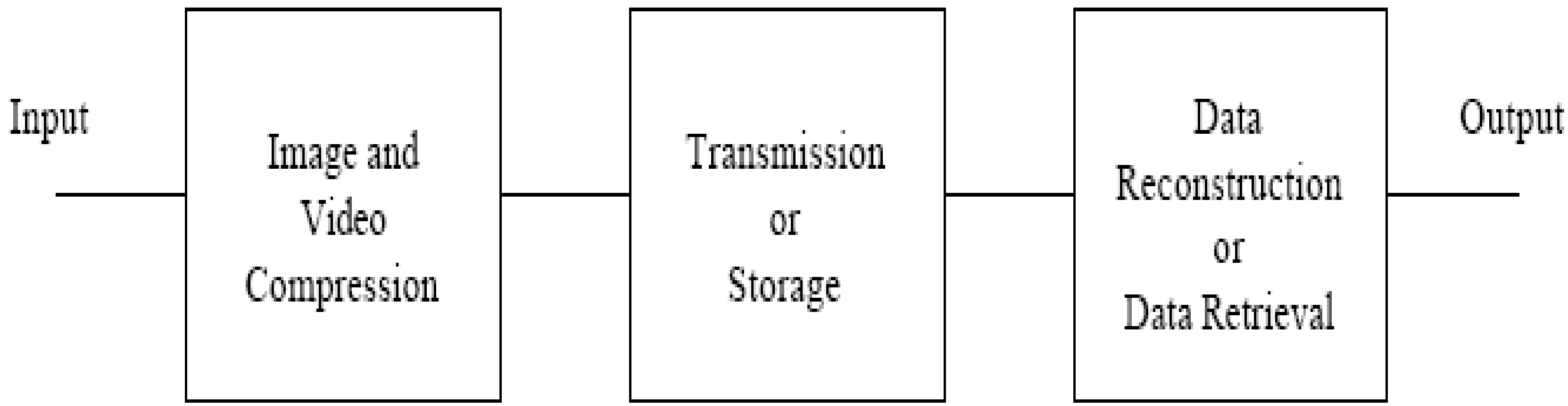
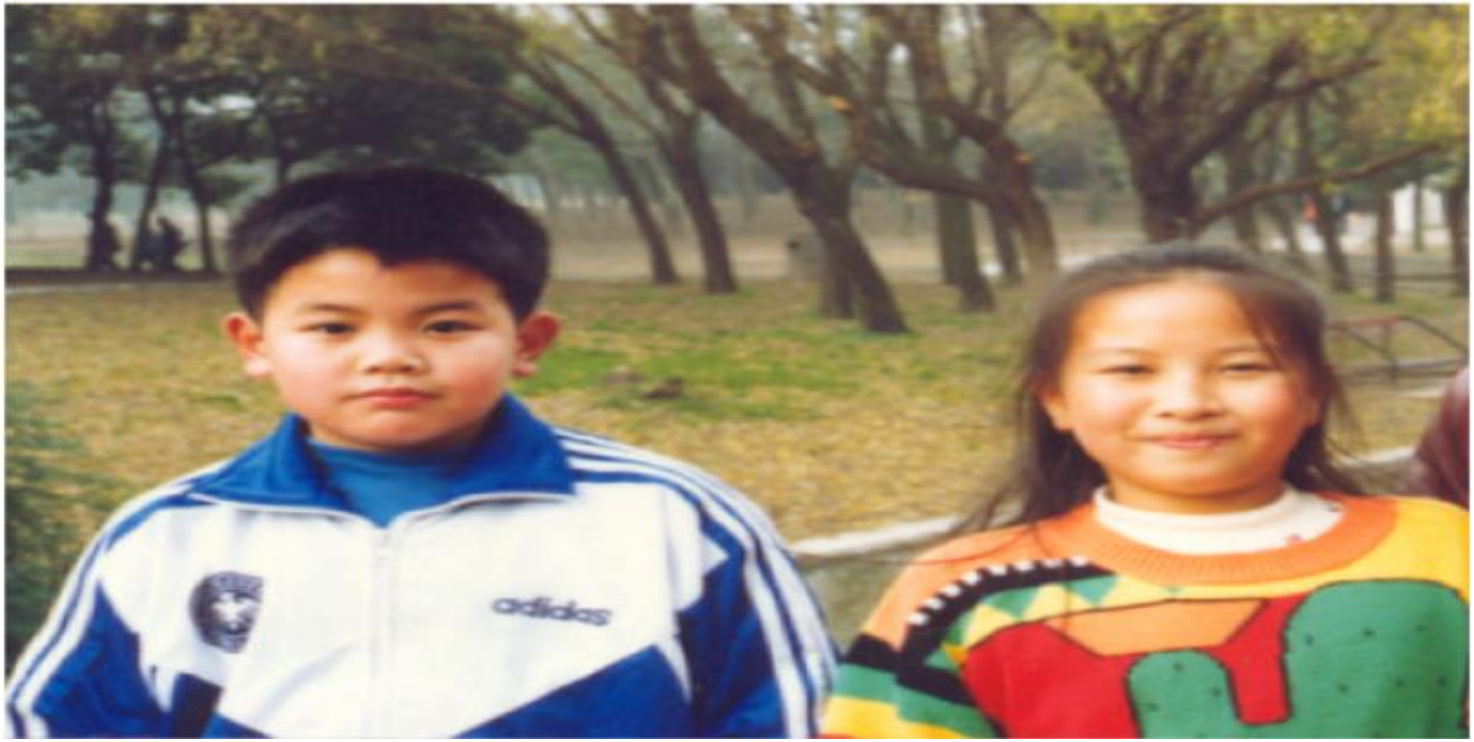


Image and video compression for visual transmission and storage

Feasibility of Image and Video Compression

- Interpixel Redundancy /Spatial Redundancy

  Statistical correlation among pixels within an image frame: intraframe redundancy.



A picture: "Boy and Girl"

Interpixel Redundancy /Spatial Redundancy

- **Not necessary to represent each pixel** in an image frame independently. Instead, **one can predict a pixel from its neighbors**.
- **Removing a large amount of the redundancy** within an image frame, we may save a lot of data in representing the frame, thus achieving data compression.

Temporal Redundancy

❖ Statistical correlation between pixels from successive frames: **Interframe Redundancy.**

(a) 21st and (b) 22nd frames of Ms. America Sequence

Temporal Redundancy

- For a videophone-like signal with moderate motion in the scene, on average, **less than 10% of pixels change their gray values** between **two consecutive frames by an amount of 1% of the peak signal**.
- Removing a large amount of temporal redundancy leads to a great deal of data compression.

Psychovisual Redundancy

- Originates from the characteristics of the human visual system (HVS).
- Visual information not perceived equally by HVS.
- If apply less data to represent less important visual information, perception will not be affected.
- In this sense, **Some Visual Information is Psychovisually redundant.**
- Eliminating psychovisual redundancy leads to data compression.

Wonders in Image Processing with ML, DL

- **Google Lens** an app launched by Google, which uses Image processing techniques along with AI technologies and Deep Machine Learning
- **Healthcare Industry-**
  - Image analysis can be of great use in the healthcare industry.
    - Medical X-ray
    - Patients
- **Defence**
- **Automobile Industry**
- **Agriculture**

Image Processing in Health Analytics

- **CT-Scan**

- MRI-Scan

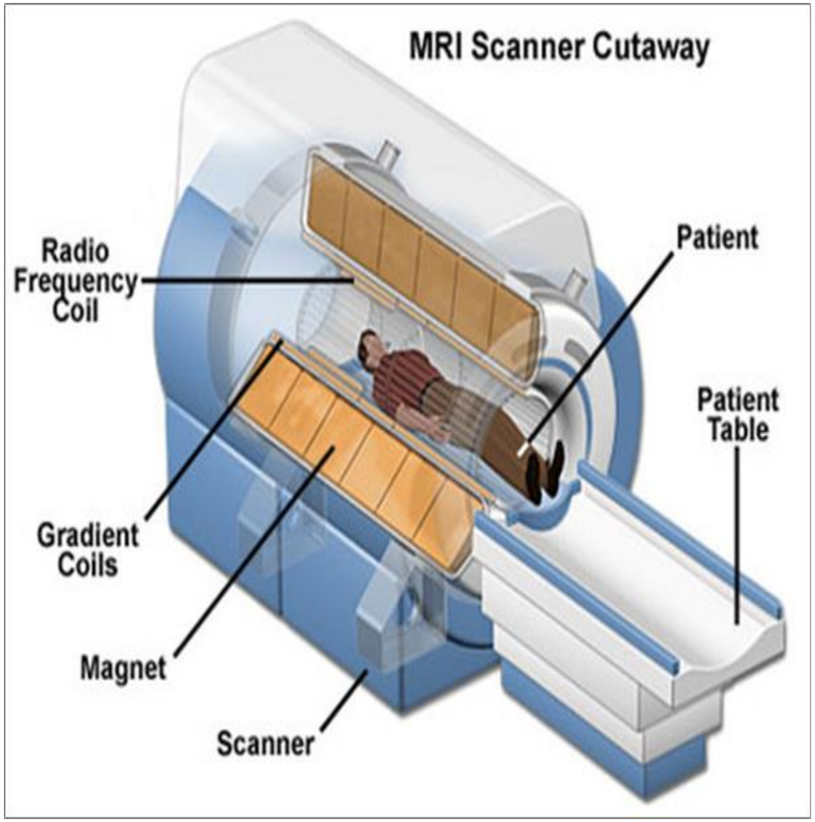


Image Processing in Health Analytics

CT Scan	MRI
Uses **Multiple X-rays**	Uses **Magnets and Radio Waves**
Doesn’t show Tendons and Ligaments	Show Tendons and Ligaments
Not as good as **MRI**	Better for examining **Spinal Cord**
Better suited to cancer, pneumonia, abnormal chest x-rays, bleeding in the brain, especially after an injury.	Better suited for nerve injuries, tumours, brain injuries, stroke, etc.
Here brain tumour is visible but not as clear as in MRI	Here brain tumour is more clearly visible
CT scans provide a better image of the Lungs and organs in the Chest cavity between the lungs.	Not good for Lungs as they are filled with air and there is a low density of the hydrogen atoms required to create MR images

Image Processing in IoT

- Smart Agriculture
- Smart Home
- Drone Based Surveillance
- Smart City etc.

Smart Agriculture

- IoT sensors and IP algorithms **to improve quality of products** using accurate measurements
- **Failure rate** can be decreased
- **Alert system** for farmers in case of any disastrous situation

Methodology

- **IoT Sensors** will continuously monitor different environmental factors
  - Temperature
  - Humidity
  - Soil moisture
  - Light intensity
- **A JPEG camera** will capture images of crops after certain interval

Fig. Block Diagram.

Smart Agriculture-Robotic Pickers

Weed Detection

Weed Detection

- **Weed detection techniques** used algorithms based on **edge detection, color detection, classification based on wavelets.**
- Real time weed recognition system for identifying outdoor plant using machine vision uses **edge based classifier to identify broad and narrow weeds.**
- Images acquired in **RGB were converted to gray scales** and used to process as binary image.
- **Bright pixels in dark background were identified as weed and classified as in broad and narrow using threshold values.**

Diolch
Kiitos
Sheun
umesc
Kasih
Todah
Mamnoon
Shnorhakalutiun
Te ekkür
Dekuju/Dekujeme
Hvala
Shokriya
Dziękuję
Shokrun
Spaas
Mul
Ačí
Gamsahapnida
Ngiyabonga
Dank
Takk
Cam
Gra
or
Xie
Kop
Salamat
Merci
al
Dhanyavaad
Dakujem
Waad
Daw
Dhanyavaadalu
Dankie
Go
Grazie
Faleminderit
Dhanyavad
Khopjai
Kruthagnathalu
krap
ThankYou
kun
Arigatou
Tack
Shukriya
Or
Dhonnobaad
ederim
Asante
Grazzi
raibh
Gracias
Nandree
Blagodariya
Gomapsupnida
Euxaristo
Kun
Hain
Dhan
Fyrir
Terima
Enkosi
Danke
dank
daa

INTESITY TRANSFORMATION

By,

Prof (Dr.) P P Ghadekar

Image Enhancement

- On the completion student will be able to understand
- Necessity of Image Enhancement.
- Spatial domain operation
  - Point processing
  - Histogram based techniques
  - Mask processing
- Frequency domain operation

What is Image Enhancement

- Processing an Image to enhance the certain features of the image.
- The result is more suitable than the original image for certain specific applications
- Processing techniques are very much problem oriented
- Best technique for enhancement of X-ray image may not be the best for enhancement of microscopic images.

Different Image Enhancement Techniques

- There are two types of Image Enhancement techniques.
- **Spatial Domain technique**
  - Work on image plane itself.
  - Direct manipulation of pixels in an image.
- **Frequency domain technique**
  - Modify Fourier Transform coefficients of an image.
  - Take inverse Fourier transform of the modified coefficients to obtain the enhanced image.

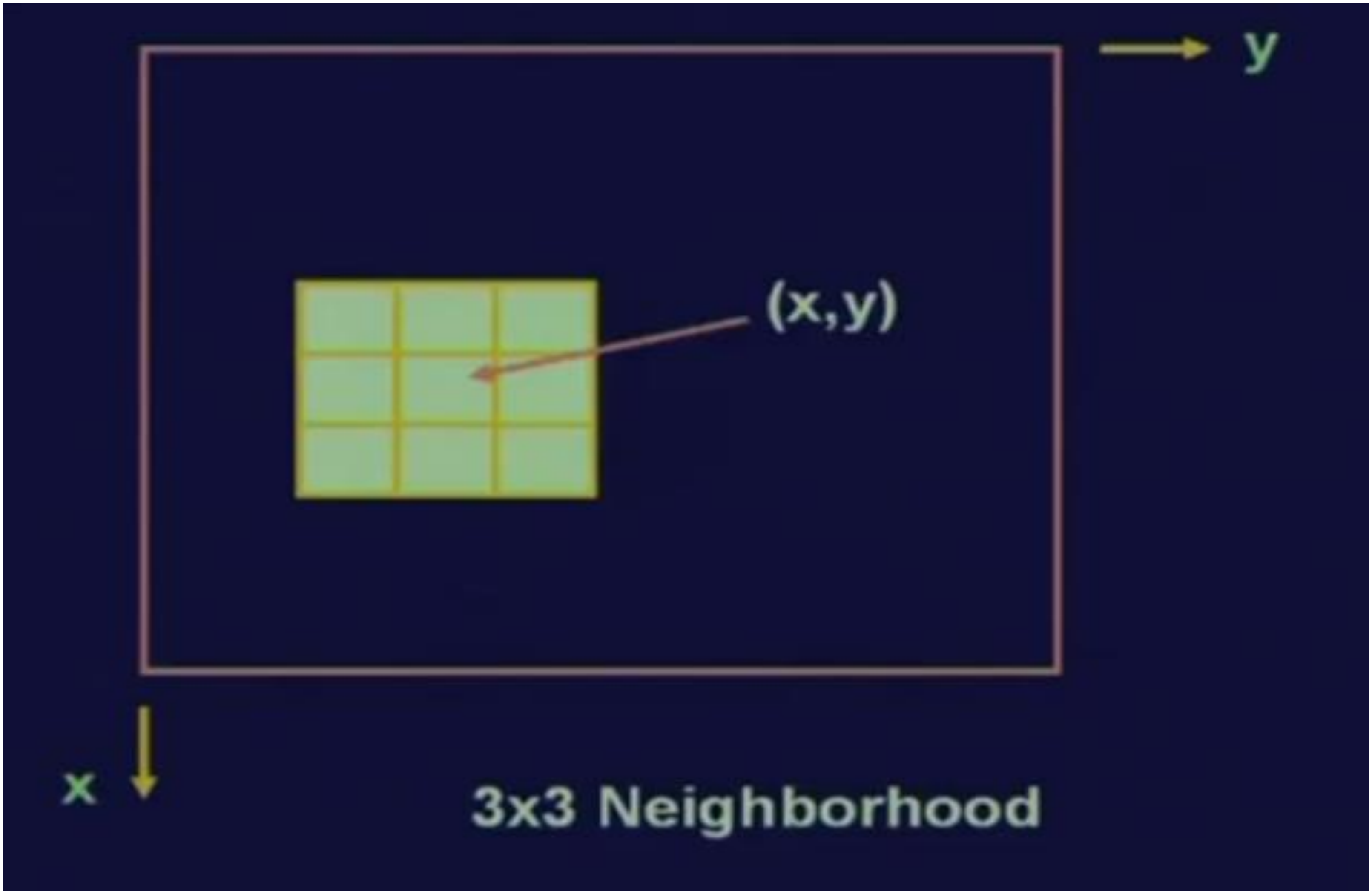
Spatial Domain technique

- Transformation function is used to transform pixels from original image to the pixels of Enhanced image.
- $g(x) = T(f(x)$
- $g(x, y) = T(f(x,y))$
- $f(x,y) \rightarrow$ original image
- $T \rightarrow$ Transformation applied on an original image.
- $g(x,y) \rightarrow$ Processed Image.

For Simplicity in notation-

$$\mathbf{S = T(r)}$$

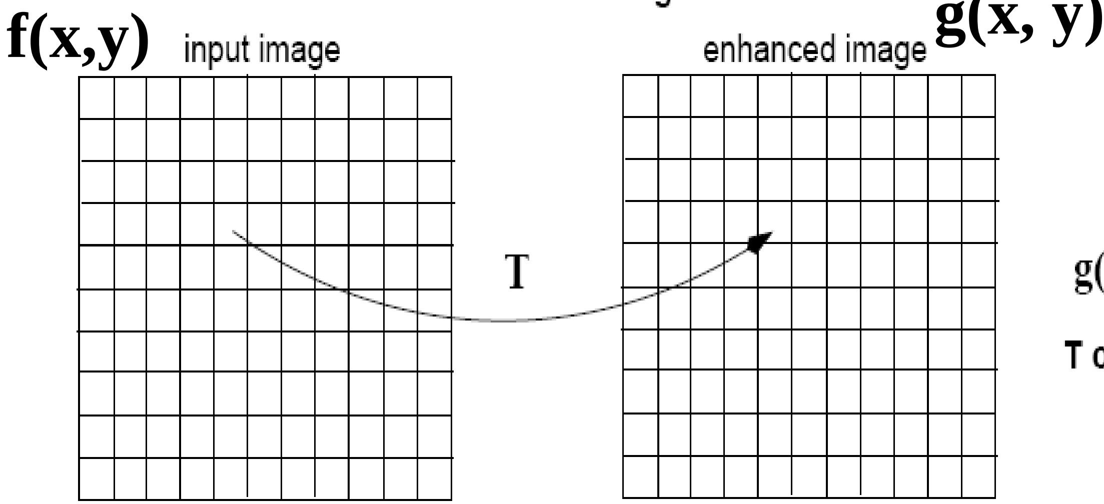
Neighbourhood



Spatial Domain Methods

Point Processing

Point Processing Methods

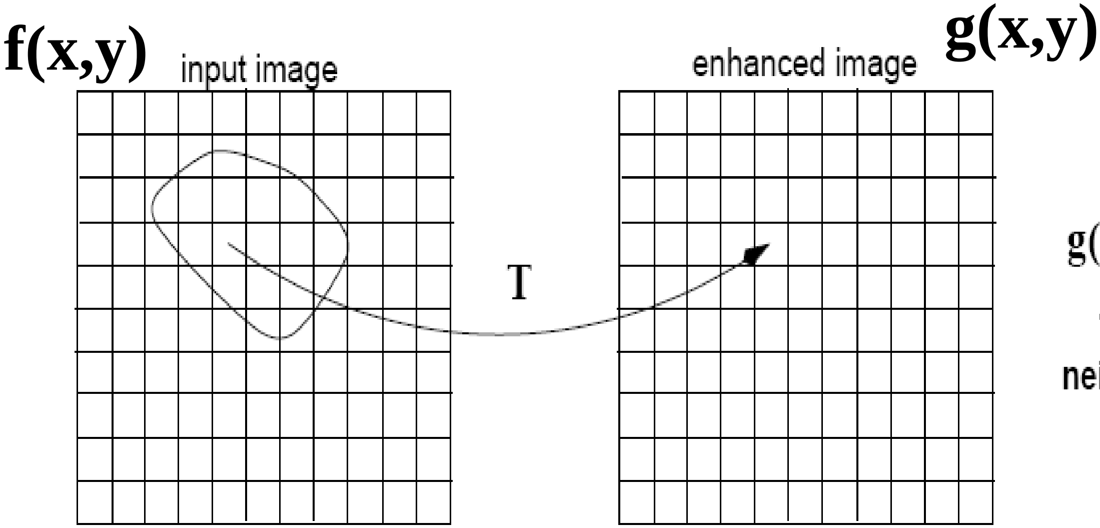


$g(x,y) = T[f(x,y)]$

T operates on 1 pixel

Area/Mask Processing

Area or Mask Processing Methods



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

T operates on a neighborhood of pixels

Types of Point operations

- Some of the examples of point operation include
  - Brightness modification.
  - Contrast manipulation.
  - Histogram Manipulation.

Brightness Modification

- Increasing the Brightness of an Image.

$$g[m,n]=f[m,n]+k$$

- Decreasing the Brightness of an Image.

$$g[m,n]=f[m,n]-k$$

Contrast Adjustment

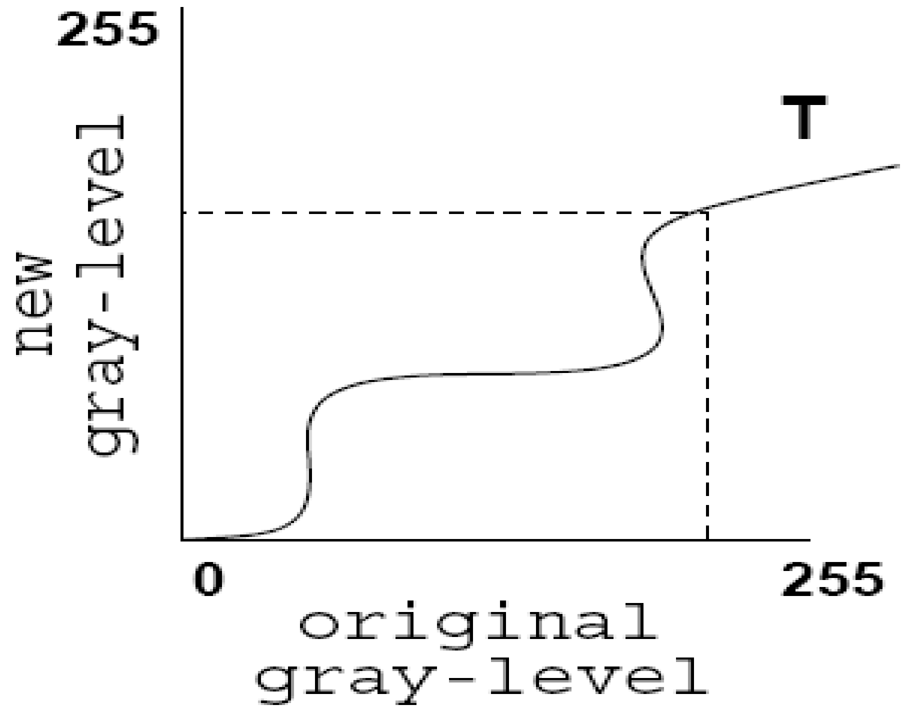
- **Luminance**, the intensity per unit area. It is amount of an energy perceived by an observer.
- **Brightness** is the Perceived luminance of the surrounding.
- **Contrast** is used to emphasise the difference in luminance of objects.
- **Contrast Adjustment** is done by scaling all pixels of the image by a constant k. It is given by-

$$g[m,n]=f[m,n]*k$$

Changing the contrast of an image, changes the range of luminance values present in the image.

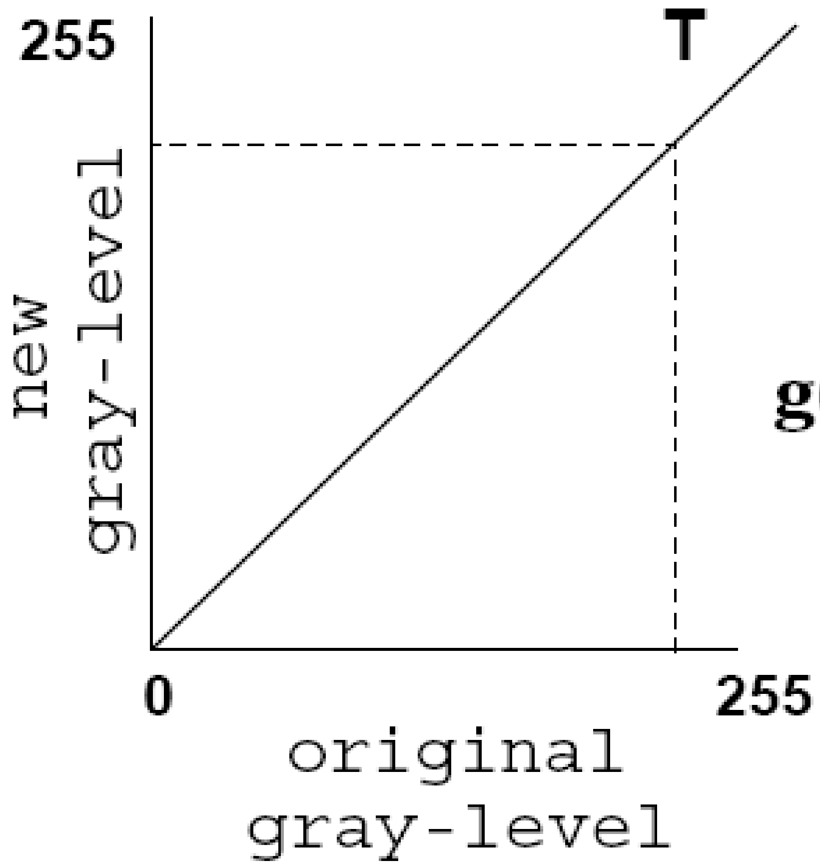
Point Processing Transformations

- Convert a given pixel value to a new pixel value based on some predefined function.



- **S=T(r)   Neighborhood size consider is 1x1**

Identity Transformation



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

Linear Gray Level Transformation

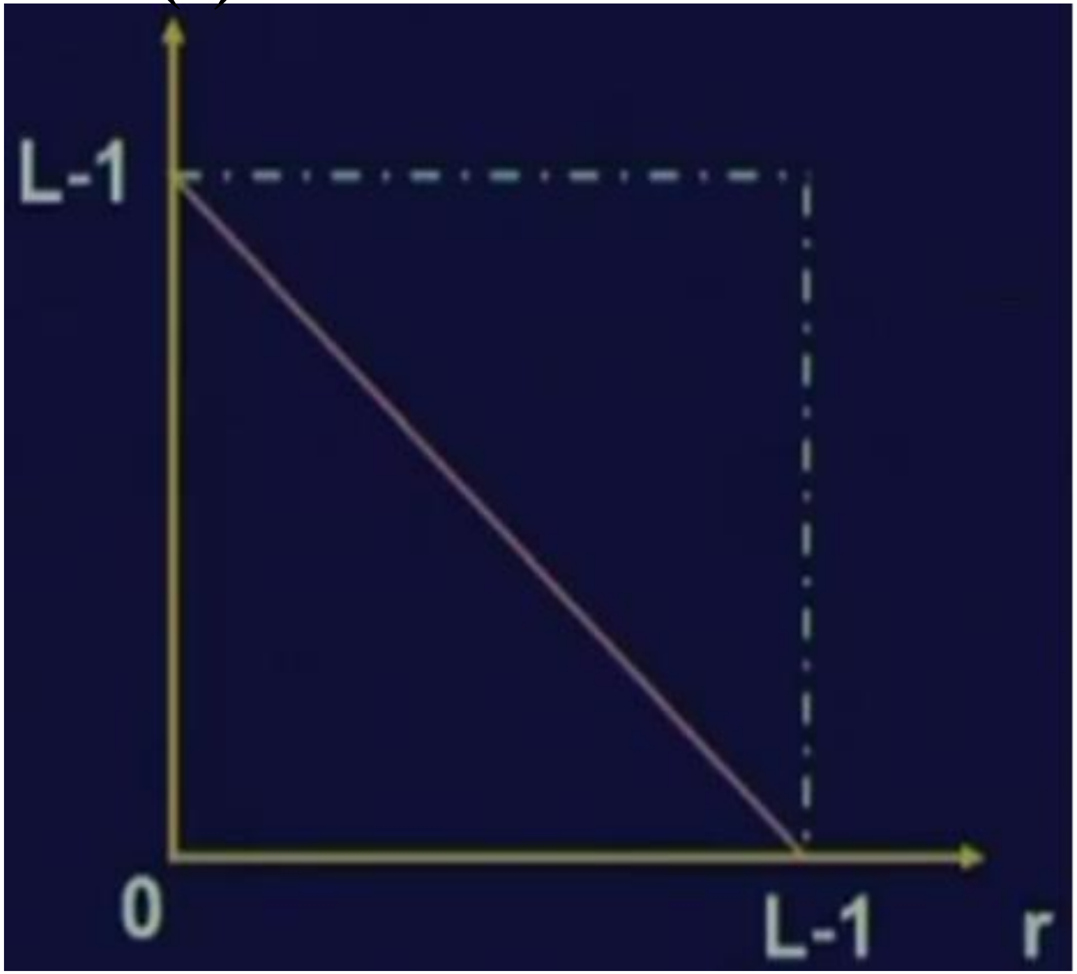
- A linear transformation of an image is a function that maps each pixel gray level value into another gray level at the same position according to a linear function.
- The linear transformation is represented by

The Linear transformation is given by

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

Image Negative

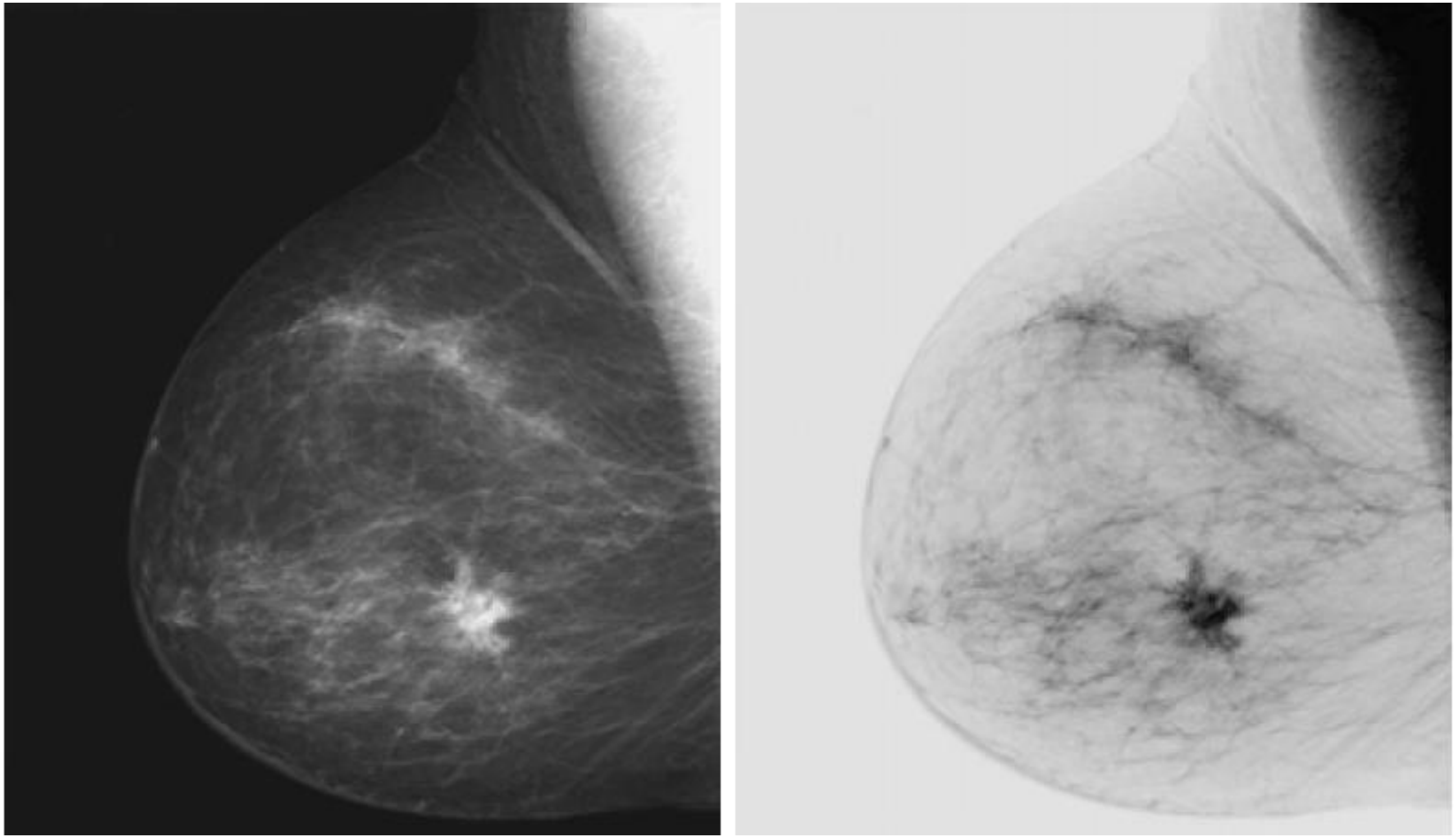
$S = T(r)$



$S = T(r) = L-1-r$

Negative Image

- **$O(r,c) = 255 - I(r,c)$**
- Mammogram Images used to detect breast Cancer

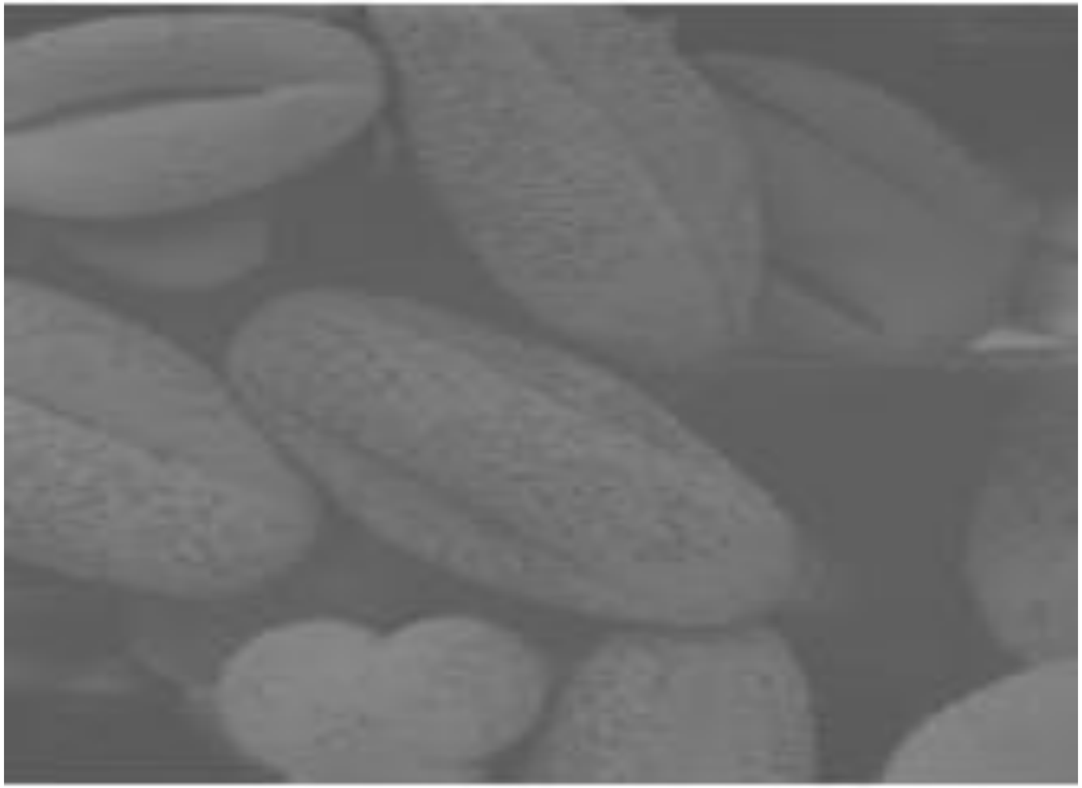


Non linear Gray level Transformation

- Non linear transformation maps small equal intervals into non-equal intervals. Some of the non-linear transformation are

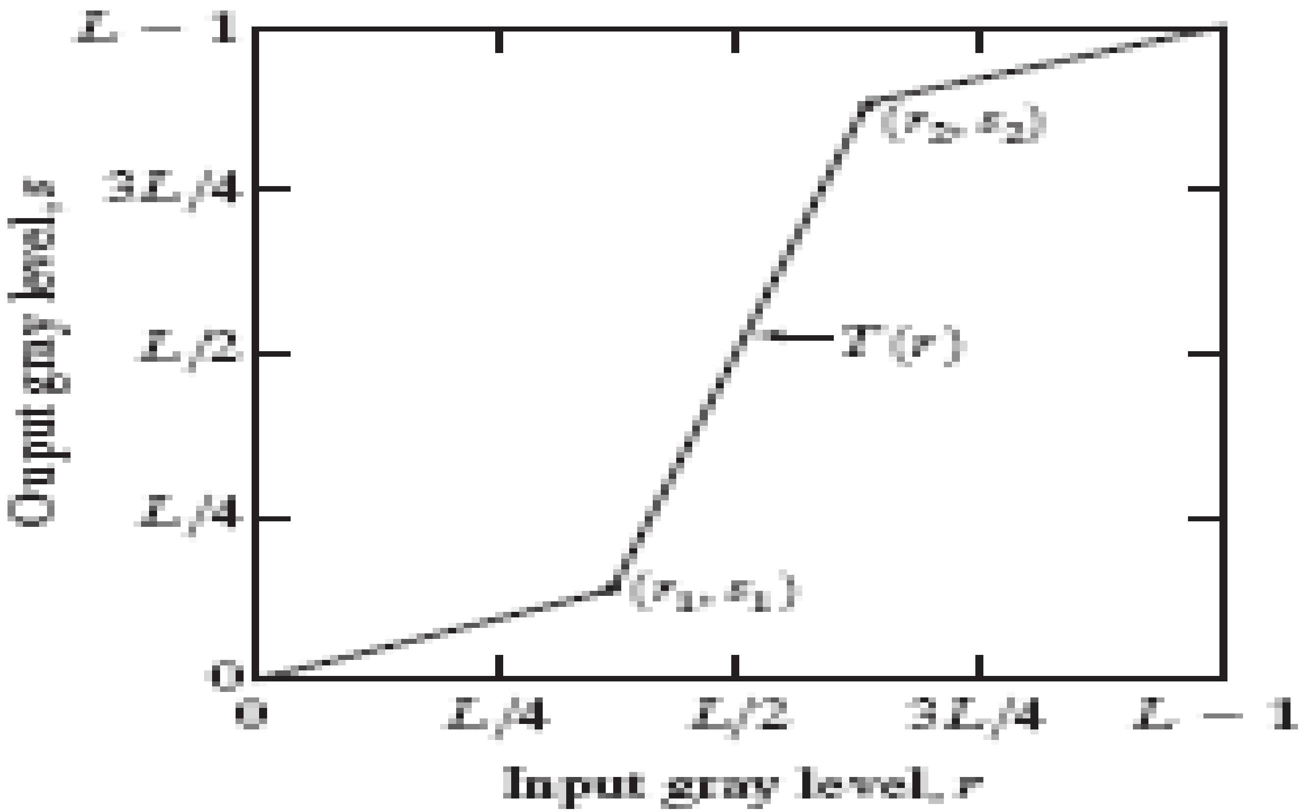
I. Thresholding
II. Gray level slicing
III. Logarithmic transformation
IV. Exponential transformation
V. Power law transformation

Contrast Stretching



Above image have very low contrast. It appears more darker.

Contrast Stretching

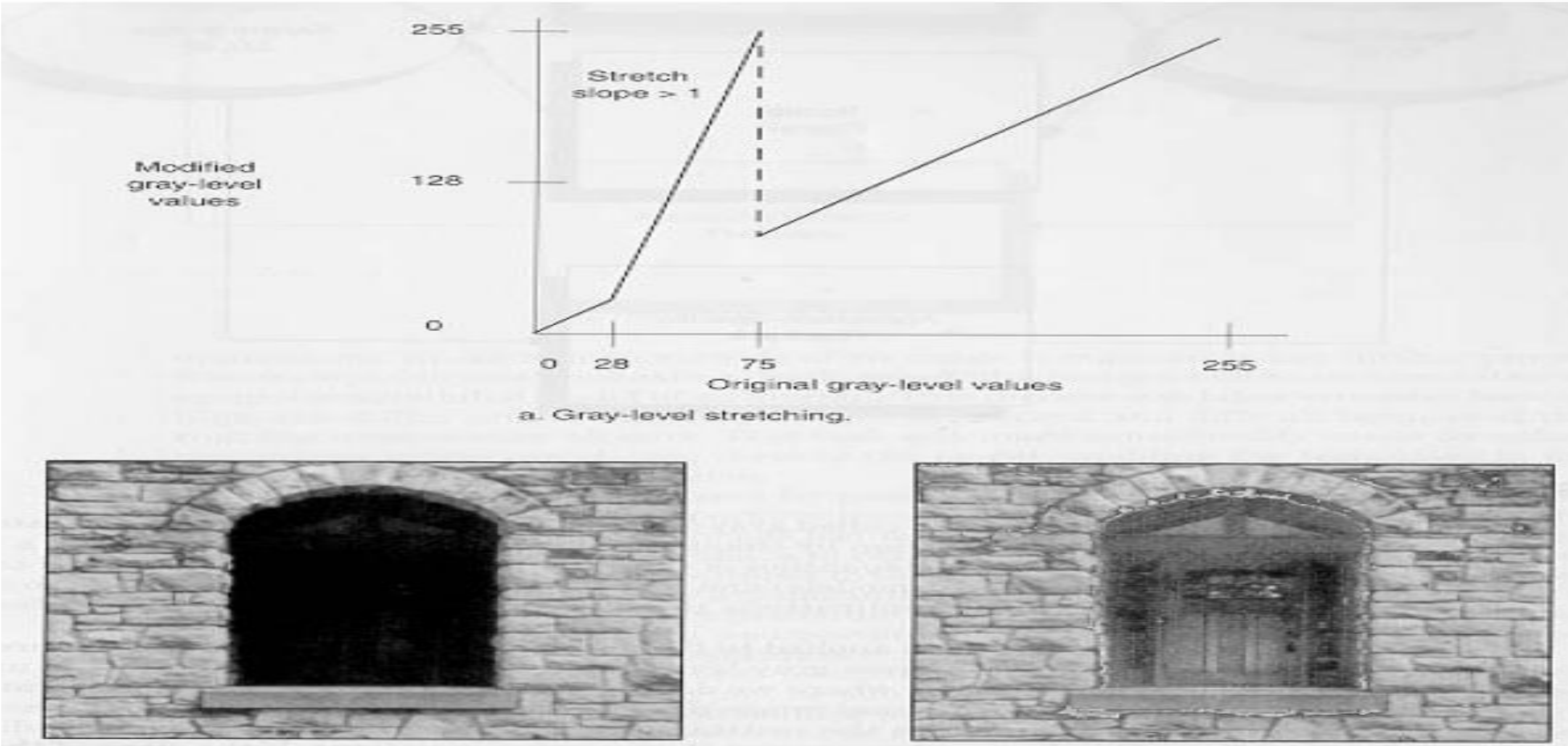


Contrast stretching. - Form of transformation function.

Contrast Stretching

(a) Result of contrast stretching. **(b) Result of Thresholding.**

Contrast Stretching or Compression

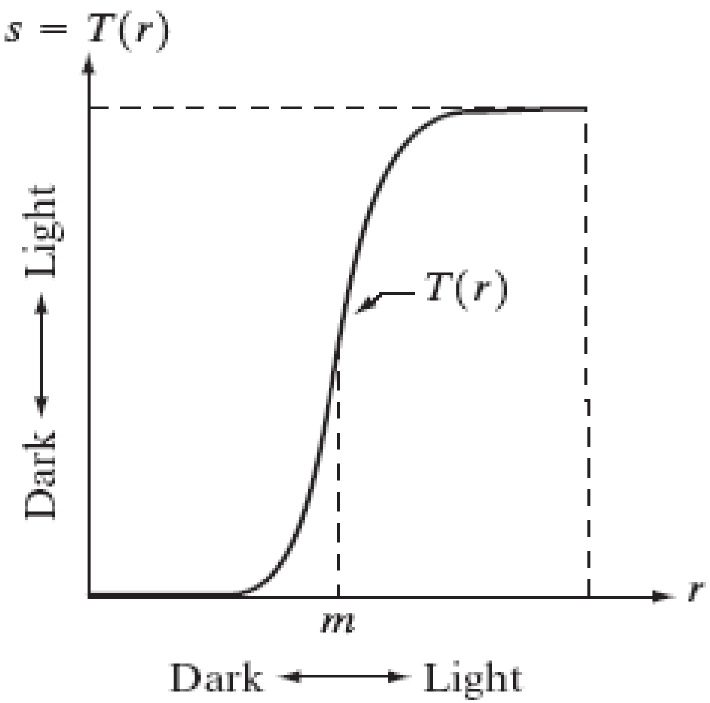


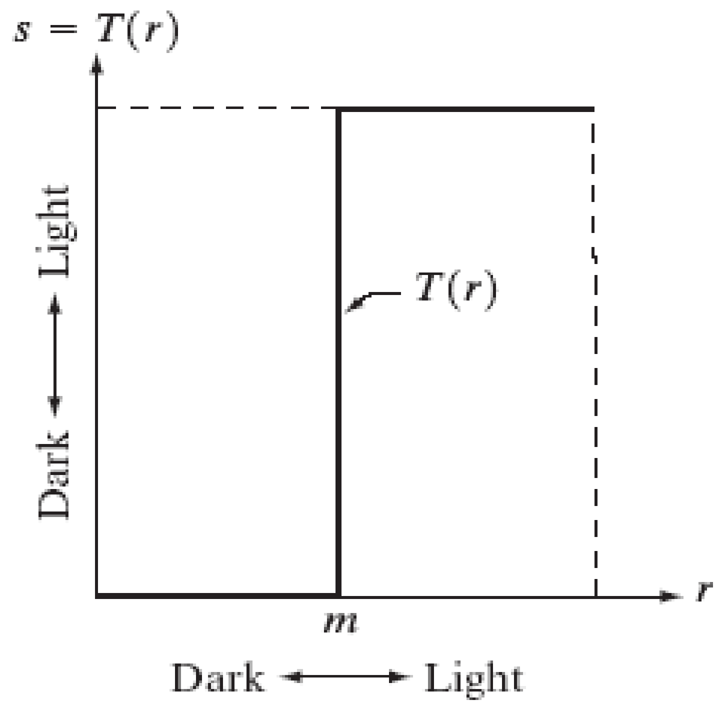
a. Gray-level stretching.

- Stretch gray-level ranges where we desire more information (slope > 1).
- Compress gray-level ranges that are of little interest (0 < slope < 1).

Thresholding

❖ **Special case of contrast compression**





Values of r below m are compressed by the transformation function into narrow range of s, toward black. The opposite effect takes place for values of r above m.

Gray level Slicing

- The purpose of gray level slicing is to highlight a specific range of gray values.
- There are two different approaches-

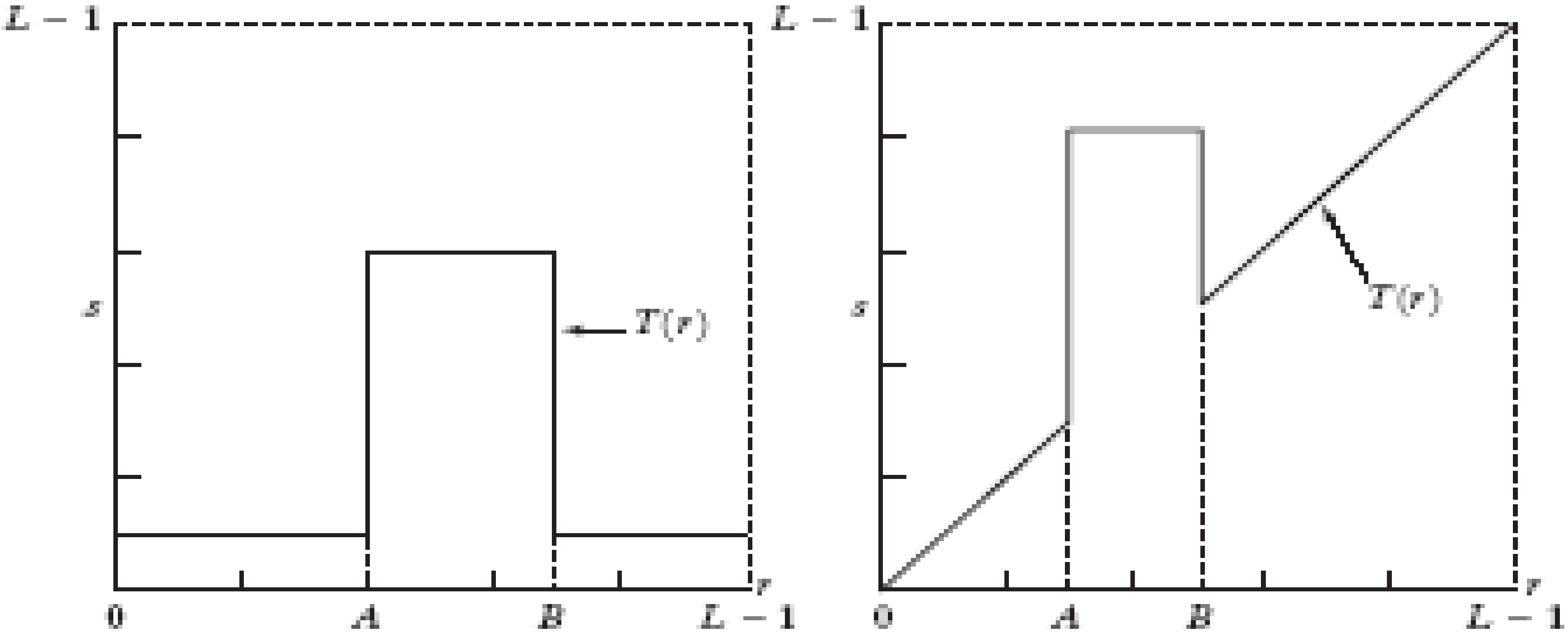
I. Gray level slicing without preserving background.

This displays high values for a range of interest and low values in other areas. The main drawback of this approach is that background information is discarded.

II. Gray level slicing with background.

This approach preserves the background of the image.

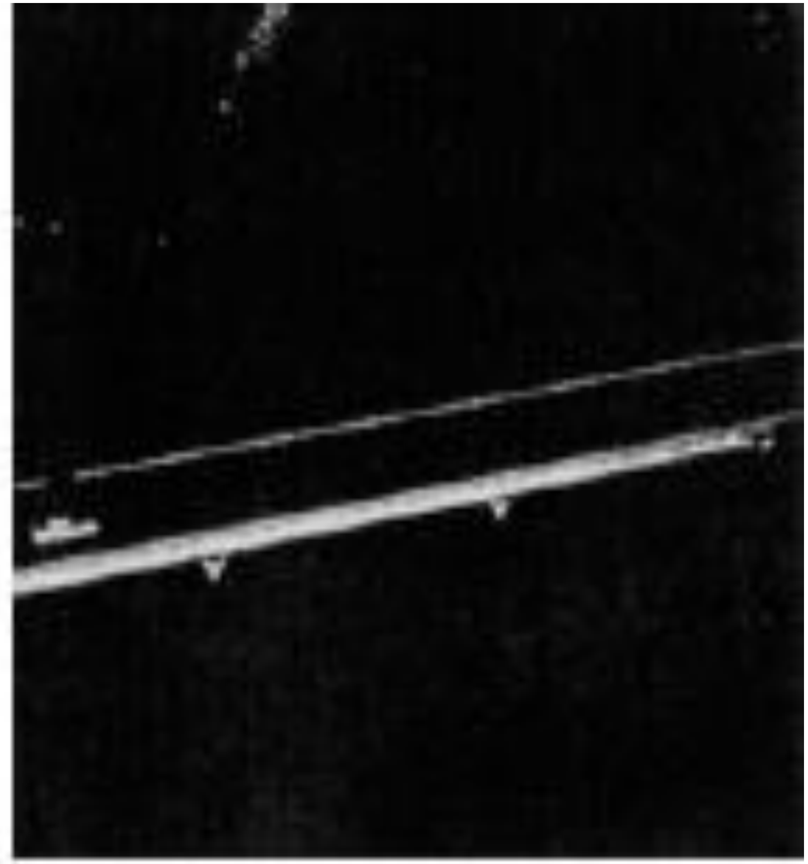
Gray Level Slicing



(a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels.

Gray Level Slicing

(a)

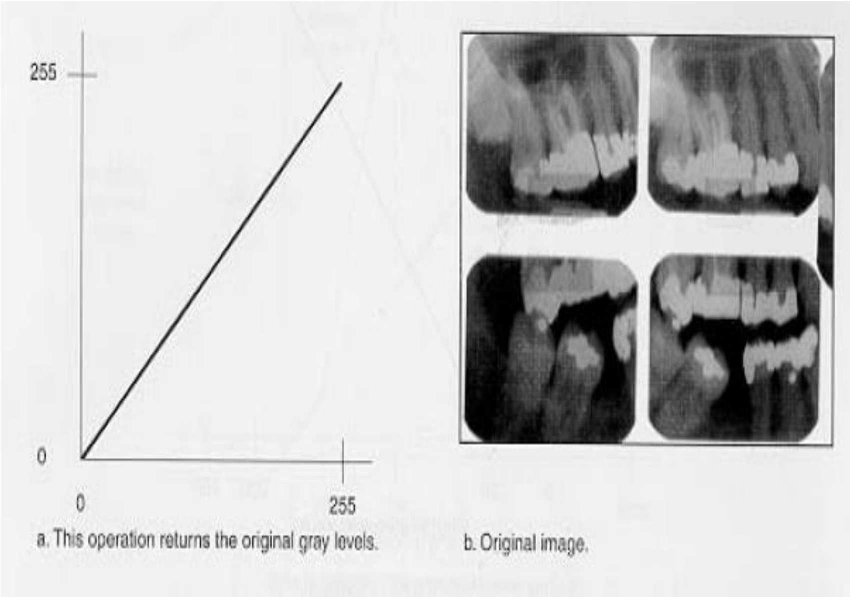


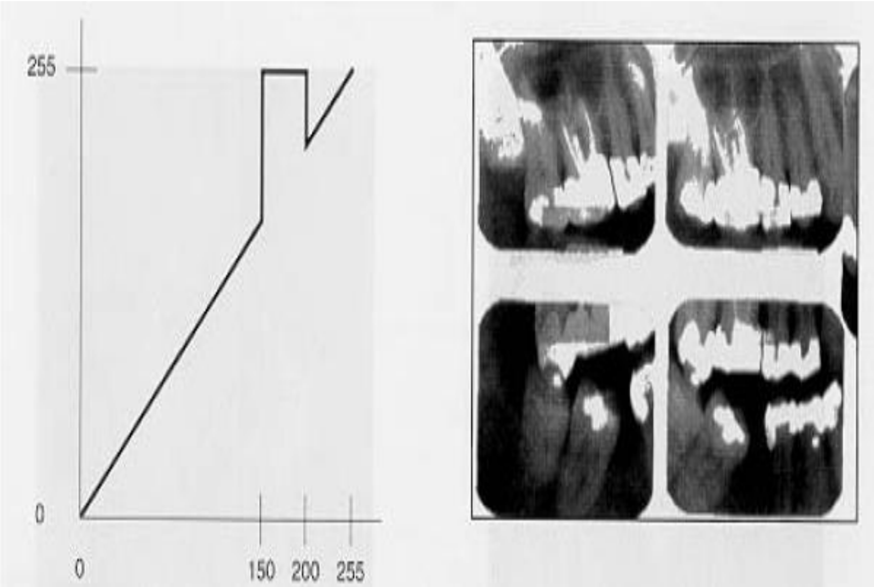
(b)

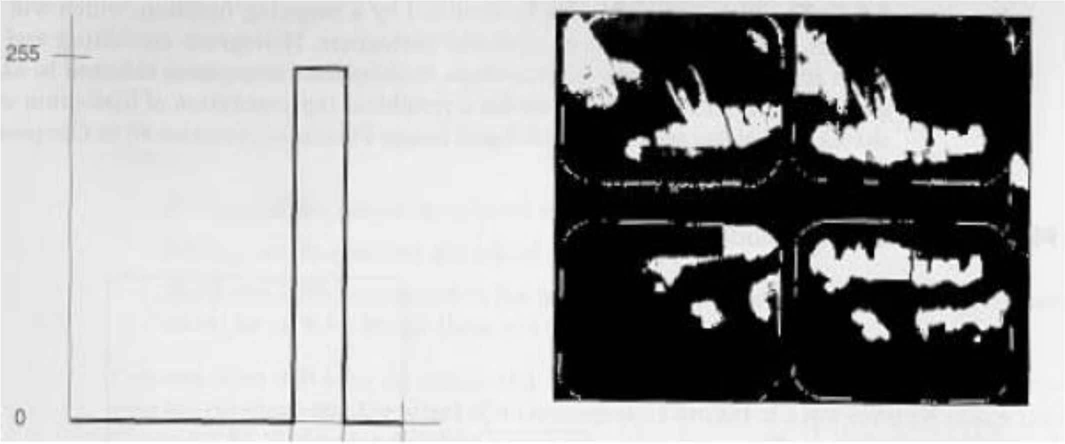
(a) An image (b) An Image using gray level slicing without background method

Gray Level Slicing

- Highlight specific ranges of gray-levels only.



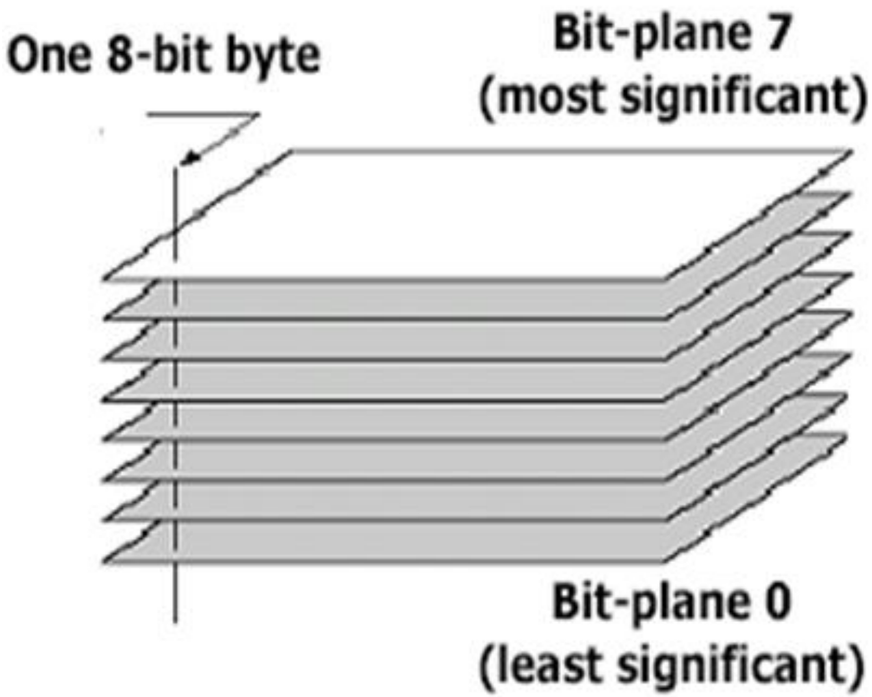




Same as double Thresholding!

Bit-Plane Slicing

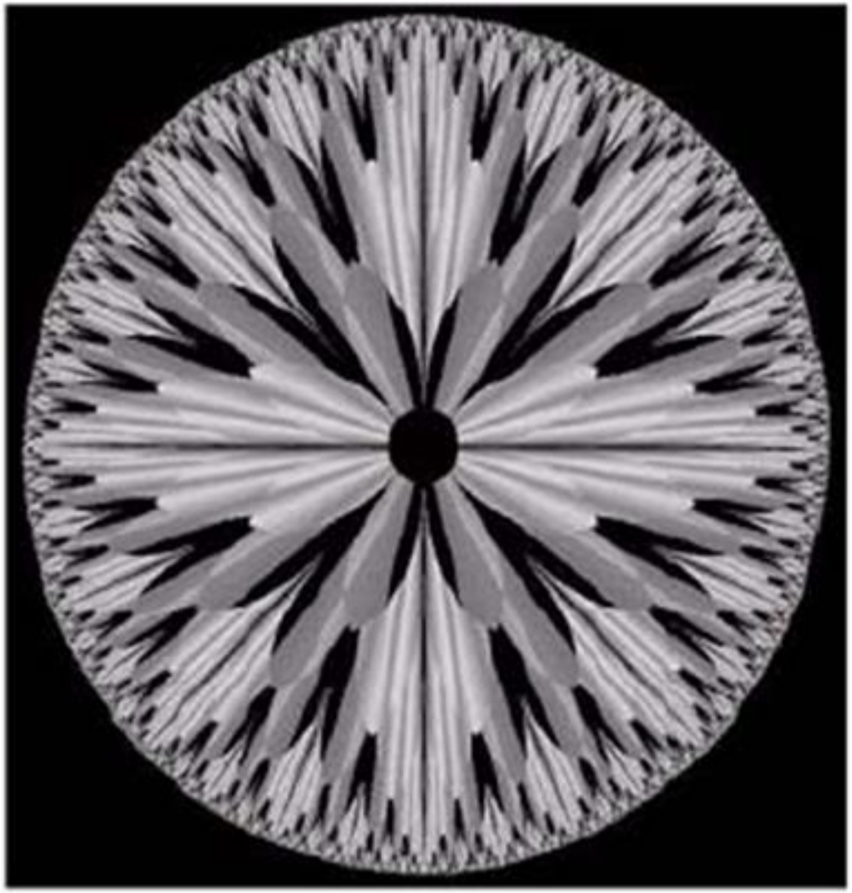
- Highlighting the contribution made by a specific bit.
- For pgm images, each pixel is represented by 8 bits.
- Each bit-plane is a binary image



Bit-Plane Slicing

- Highlighting the contribution made to total image appearance by specific bits.
- Suppose each pixel is represented by 8 bits.
- Higher order bits contains the majority of the visually significant data.
- Useful for analysing importance played by each bit of the image.

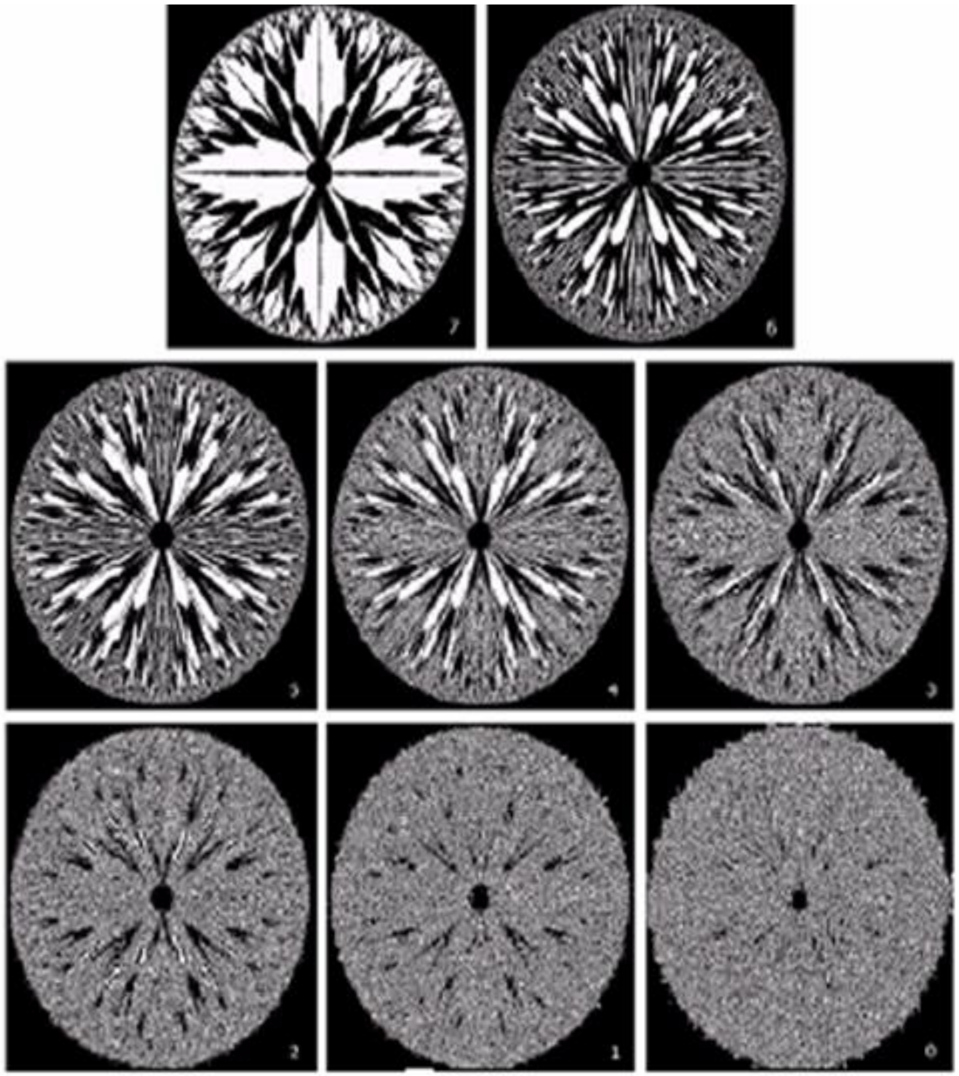
Example



An 8-bit fractal image

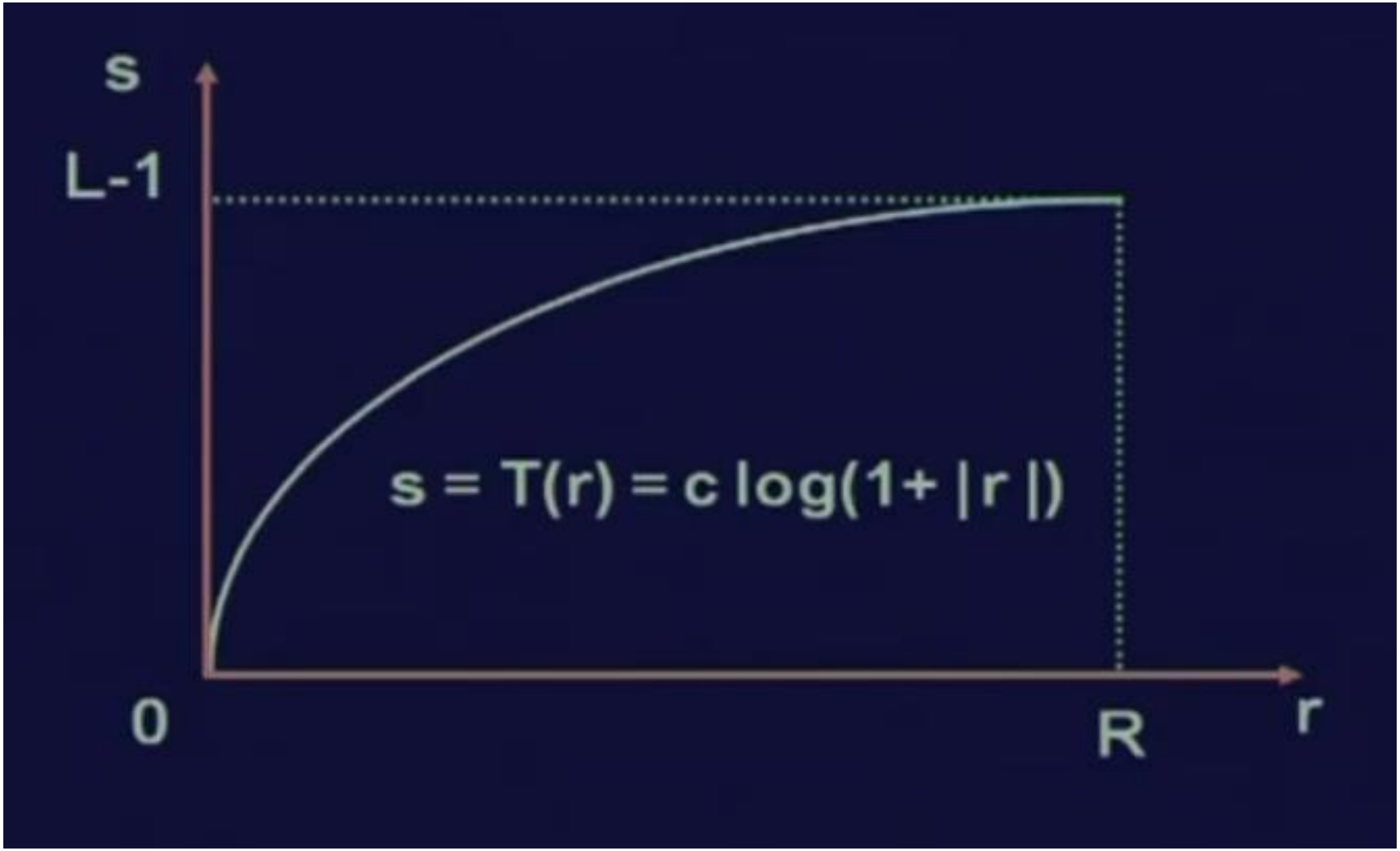
- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.
  - Map all levels between 0 and 127 to 0
  - Map all levels between 129 and 255 to 255

Example



<table>
<tr><td colspan="3">Bit-plane 7</td><td colspan="3">Bit-plane 6</td></tr>
<tr><td colspan="2">Bit-plane 5</td><td colspan="2">Bit-plane 4</td><td colspan="2">Bit-plane 3</td></tr>
<tr><td colspan="2">Bit-plane 2</td><td colspan="2">Bit-plane 1</td><td colspan="2">Bit-plane 0</td></tr>
</table>

Logarithmic Transformation



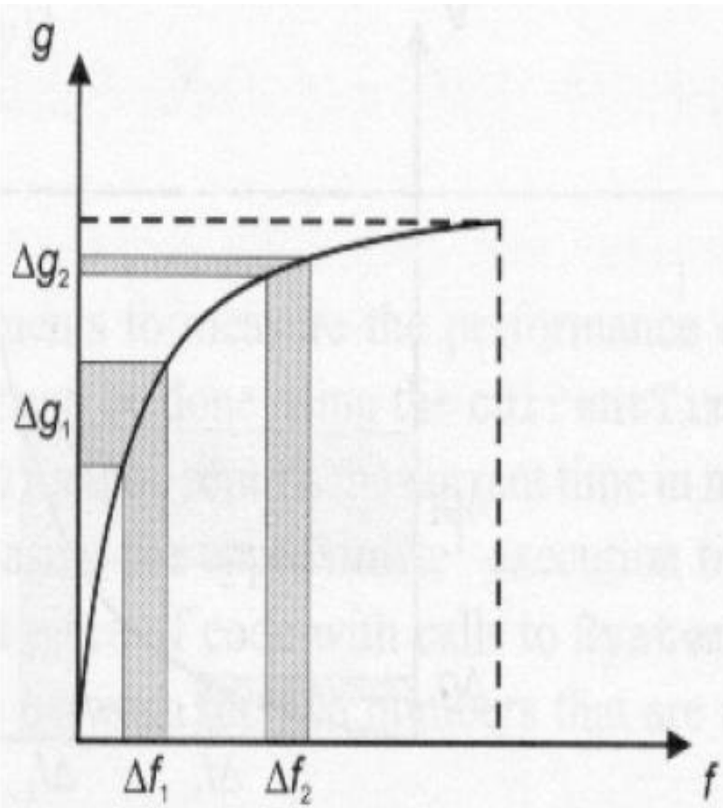
Logarithmic Transformation

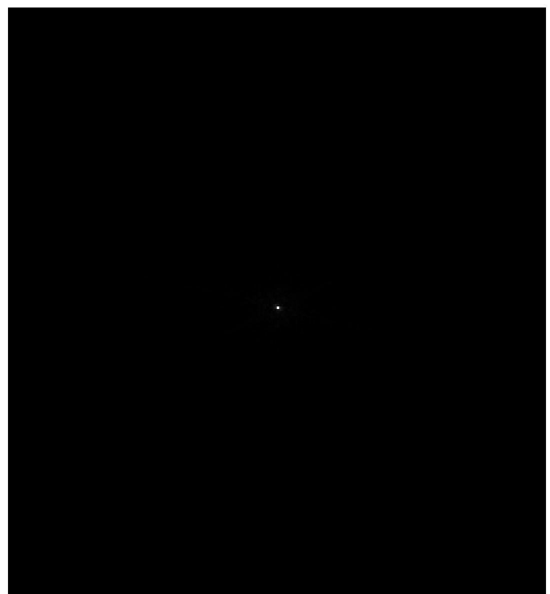
- It compresses the dynamic range of images with large variations in pixel values.
- Example of image with dynamic range: Fourier spectrum image.
- It can have intensity range from 0 to 10^6 or higher.
- We cant's see the significant degree of detail as it will be lost in the display.

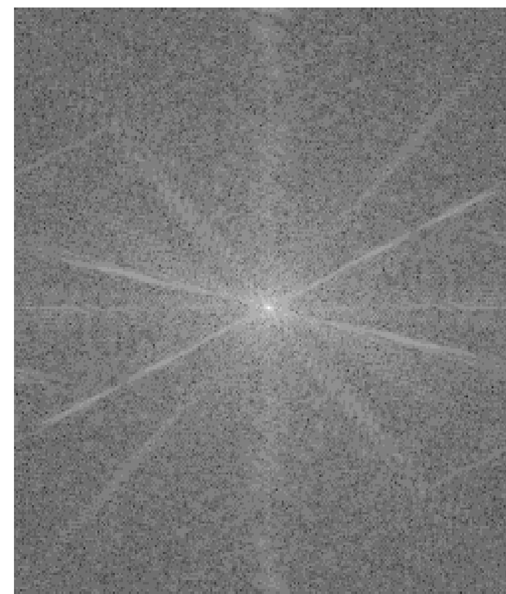
Logarithmic transformation

- Enhance details in the darker regions of an image at the expense of detail in brighter regions.

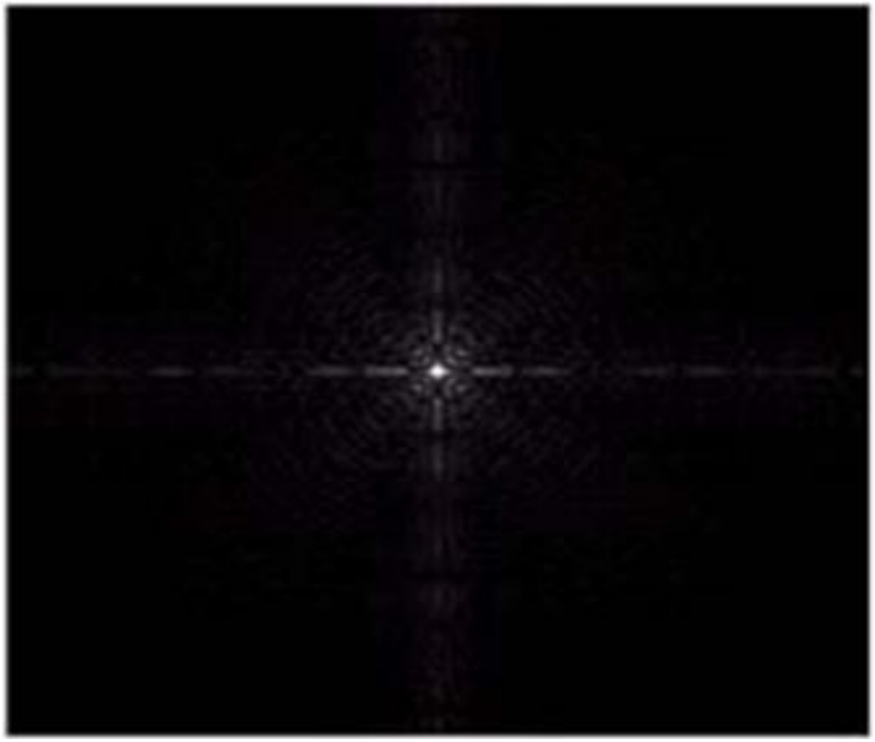
$$s = T(r) = c\log(1+|r|)$$

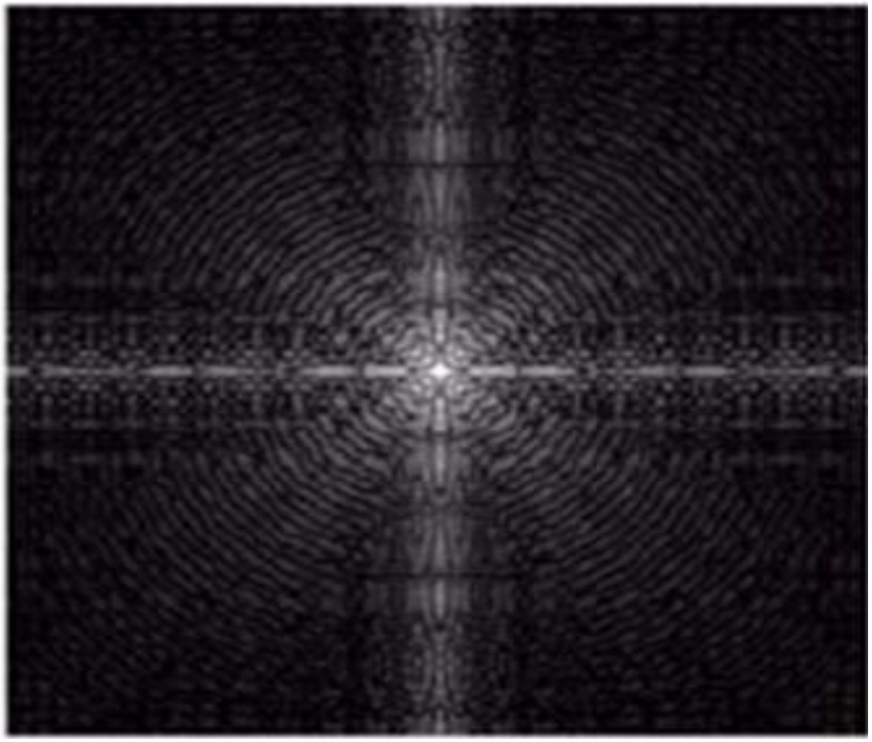






Example of Logarithmic Range





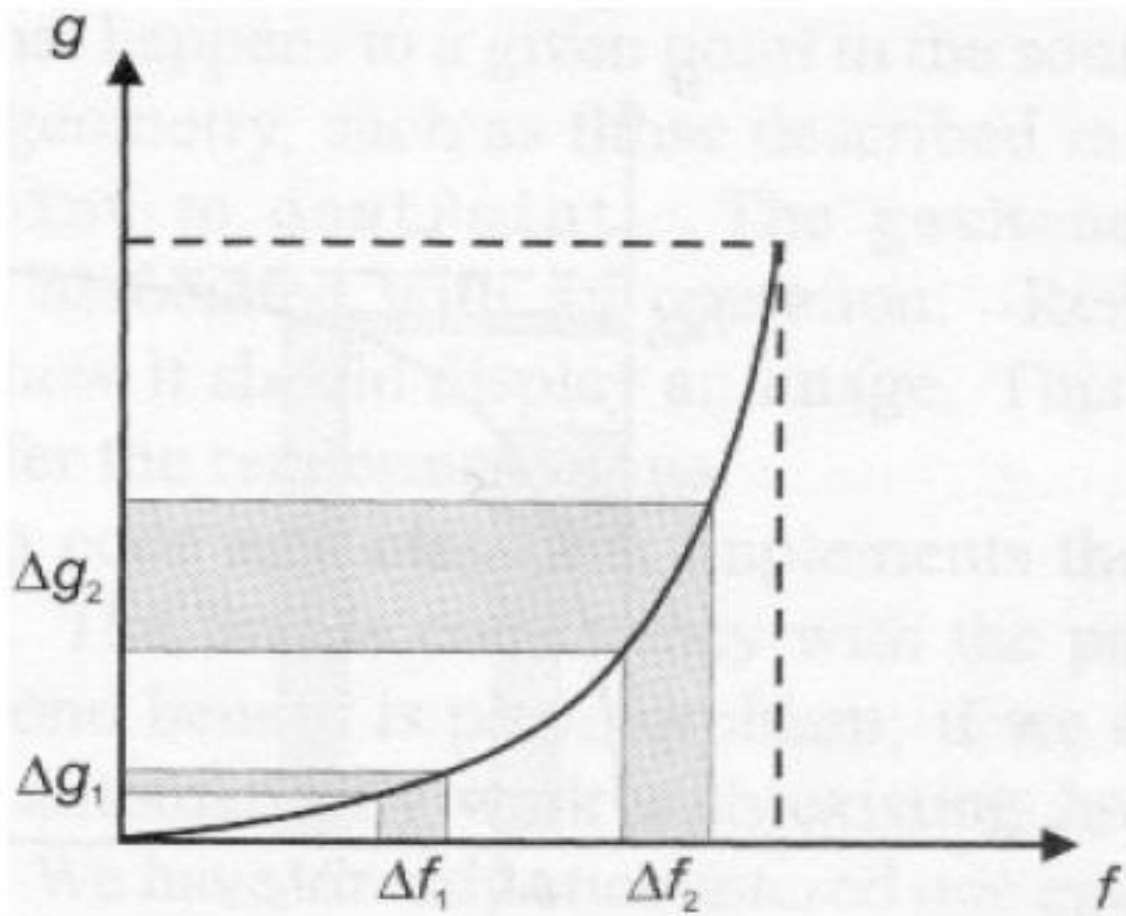
Fourier Spectrum with range=0 to 1.5 x 10^6

Result after apply the log transformation with c=1, Range =0 to 6.2

Exponential transformation

❖Reverse effect of that obtained using logarithmic mapping.

The effect of an exponential transfer function on edges in an image is to compress low contrast edges while expanding high contrast edges.



Power law Transformation

- Power law transformation is used for different imaging devices. It is used for image capturing devices. It is used for image printer etc.

$$S = T(r) = C\, r^{\gamma}$$

C-Scaling Factor

ϒ –Gamma Correction

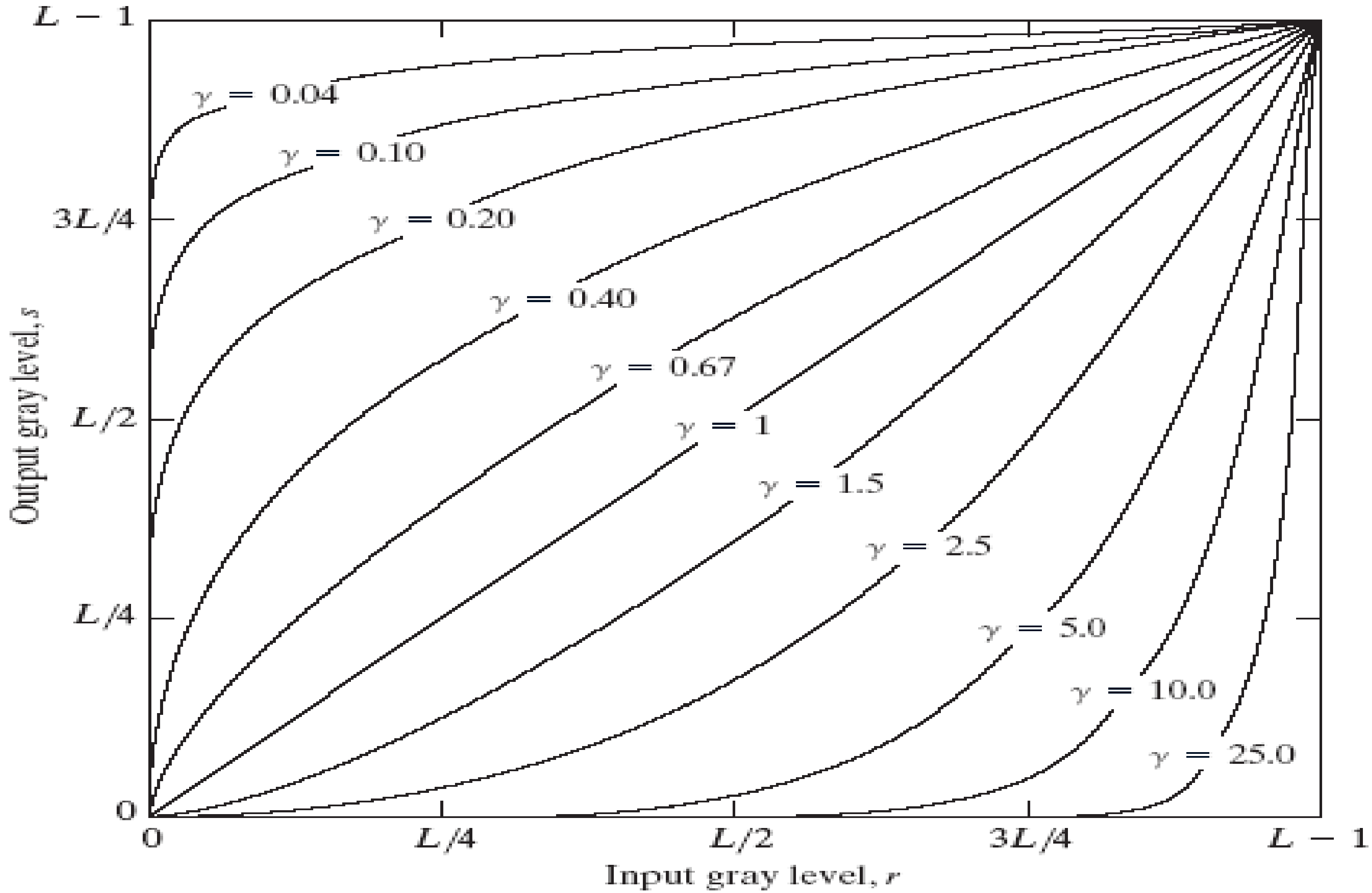
Such method is also called as Gamma Correction.

It is used for many CRT Display devices, Printing devices and Capturing Devices.

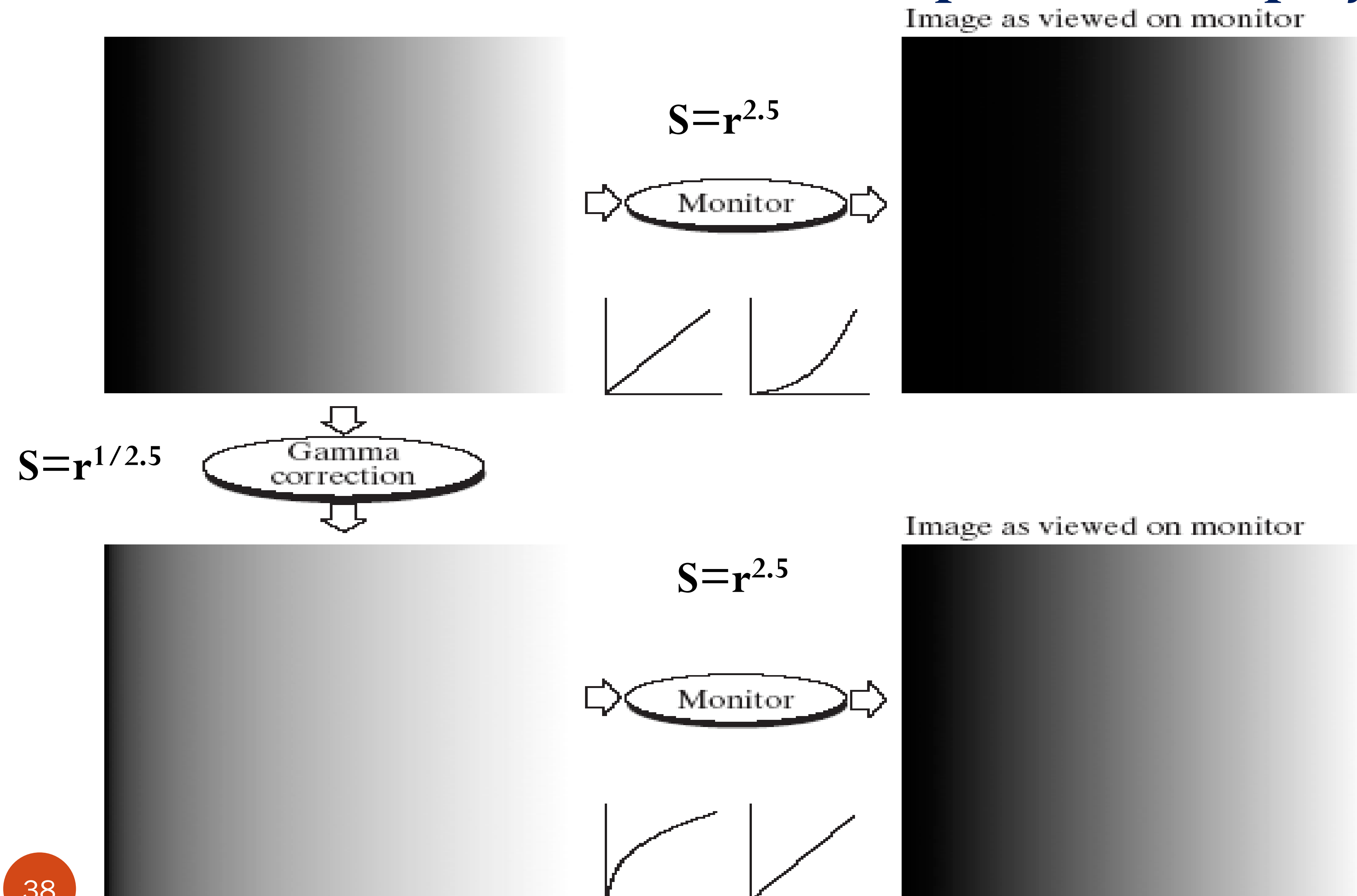
Power law Transformation

- If $\Upsilon < 1$ maps a narrow range of dark input values into a wider range of output values, while $\Upsilon > 1$ maps a narrow range of bright input values into a wider range of output values

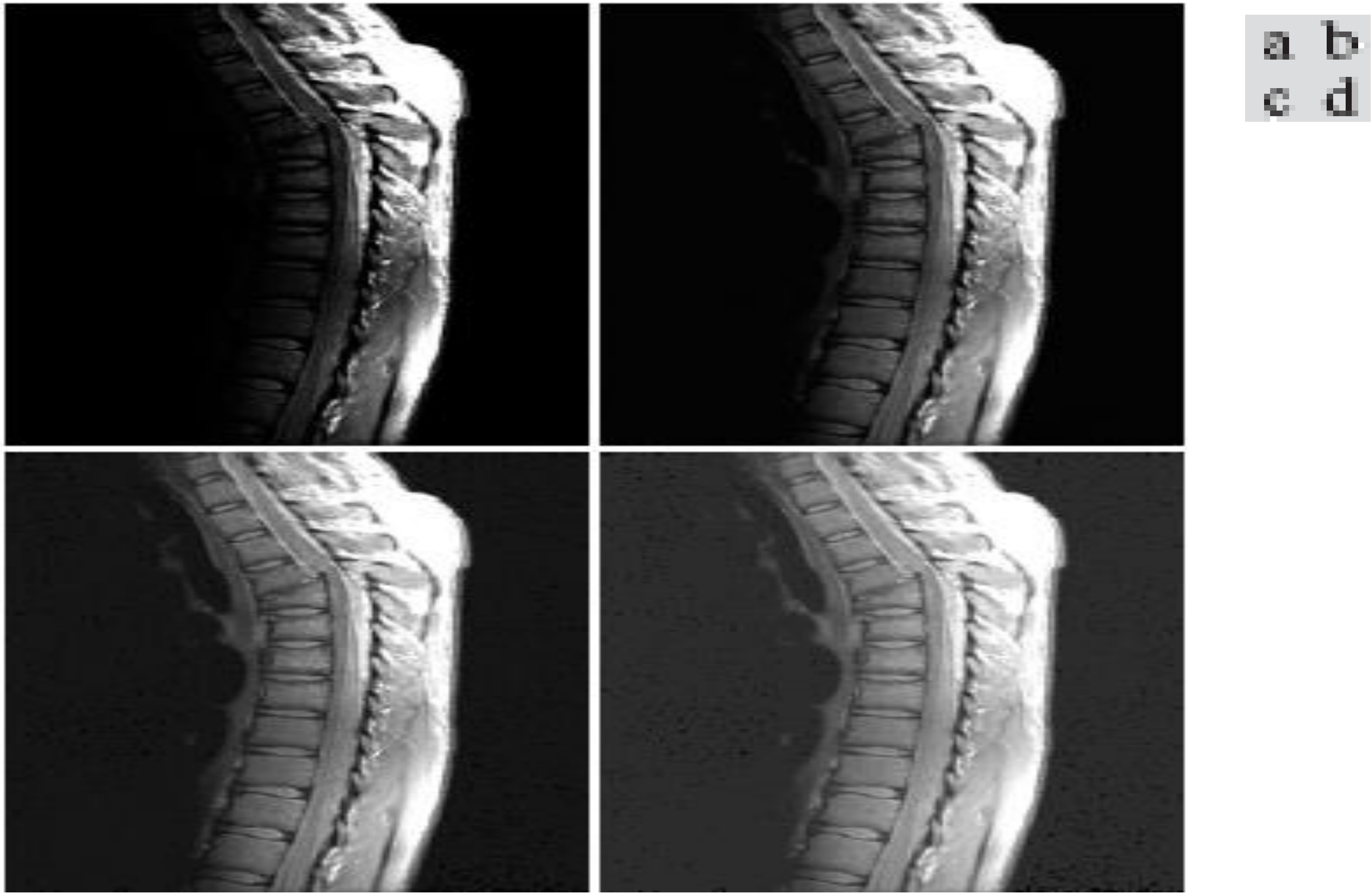
Power law Transformation



Power law Transformation-Example-CRT Display



Power law Transformation



(a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation with c=1 and g=0.6, 0.4, and 0.3, respectively.

Example –Gamma Correction

(a) Aerial image. (b)–(d) Results of applying the transformation with c=1 and g=3.0, 4.0, and 5.0, respectively.

Example –Gamma Correction

gamma=1, 0.7, 04. 0.1

Image Enhancement

- On completion the Student will be able to understand
  - What is Histogram
  - Histogram equalization
  - Histogram Specification
  - Image Subtraction
  - Image Averaging

Image Histograms

- An image histogram is a plot of the gray-level frequencies (i.e., the number of pixels in the image that have that gray level).

0	0	1	0	2	0
1	0	7	7	7	0
0	7	0	0	7	0
1	0	0	7	2	0
0	0	7	1	0	1
1	0	7	7	7	0

frequencies

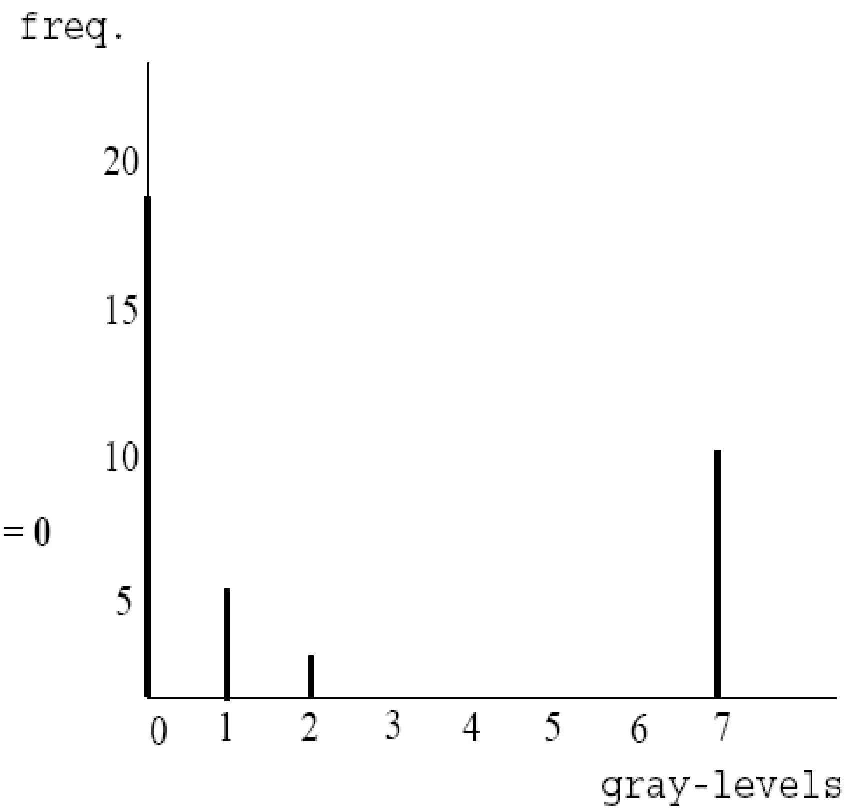
$f(0)= 18$

$f(1)= 6$

$f(2)= 2$

$f(3)=f(4)=f(5)=f(6) = 0$

$f(7) = 10$



Basics of Histogram

- The histogram of an image is a plot of the number of occurrences of gray levels in the image against gray level values.
- Histograms are the basis for numerous spatial domain processing techniques.
- It plays an important role in enhancement of perceived brightness and contrast of an image.
- It specifies the number of pixels having each gray level, but gives no hint as to where those pixels are located within the image.
- The histogram of an $N \times M$ image is defined as the percentage of pixels within the image at a given gray level:

Basics of Histogram

$$h_i = \frac{n_i}{MN} \qquad for\ 0 \le i \le G_{\max}$$

- where n_i is the number of pixels at gray level i,
- NM is the total number of pixels within the image and
- G_{max} is the maximum gray level value of the image.
- hi is the probability of gray level i

Basics of Histogram

Consider an Image have gray level intensity range [0,L-1]

r_k be the k^{th} intensity level.

Histogram $h(r_k) = n_k$

n_k is the number of pixels in the image having intensity r_k

Normalized Histogram

$$\mathbf{P(r_k) = n_k / MxN}$$

$P(r_k)$-Tells you what is the probability of occurrence of pixels having intensity value r_k

Properties of Histogram.

- The histogram is unique for any particular image, but the reverse is not true.
- The sum of each normalized histogram value over the range of gray levels present within an image equals 1.

$$\sum_{i=0}^{G_{max}} h_i = 1$$

- The average gray level of an image in terms of histogram is:

$$avg = \sum_{i=0}^{G_{max}} i \cdot h_i$$

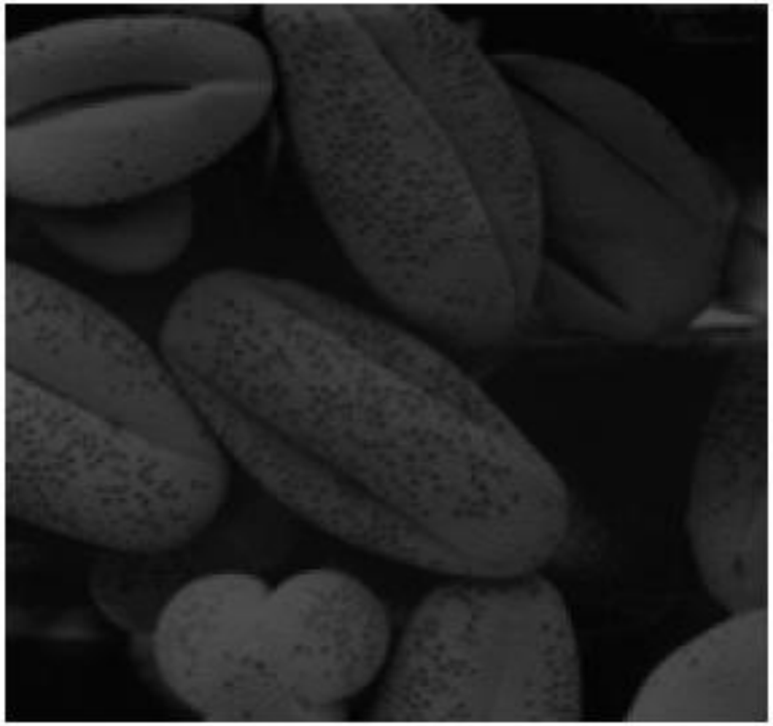
Properties of Histogram.

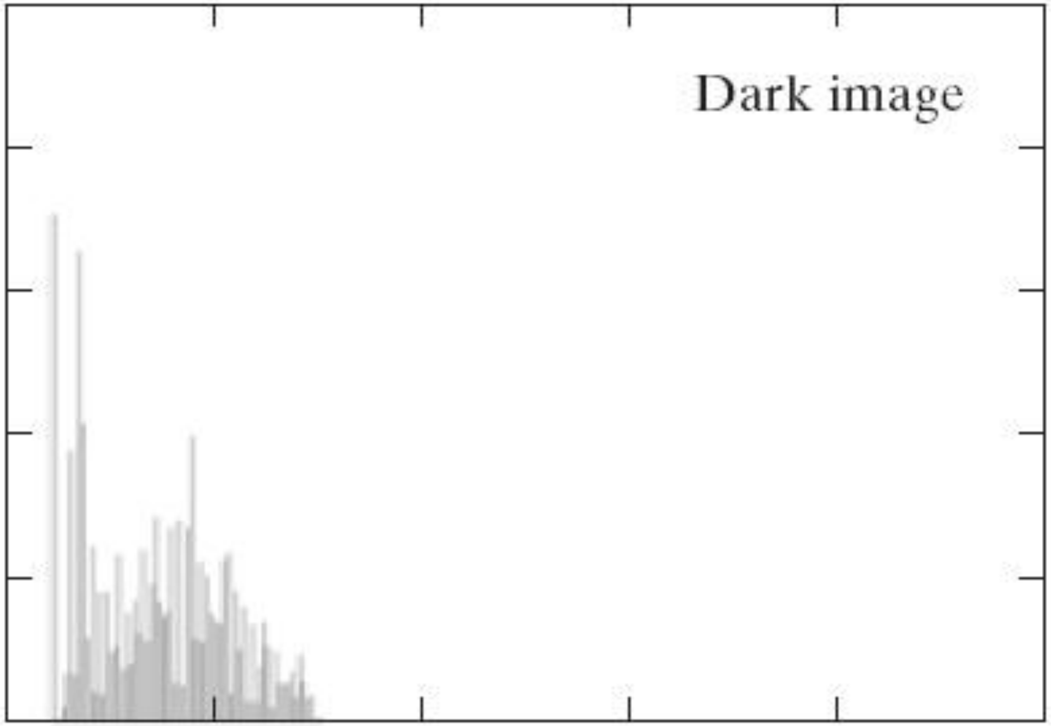
- Standard deviation of the gray levels of an image in terms of its histogram is given as:

$$std = \sqrt{\sum_{i=0}^{G_{max}} i^2 \cdot h_i - avg^2}$$

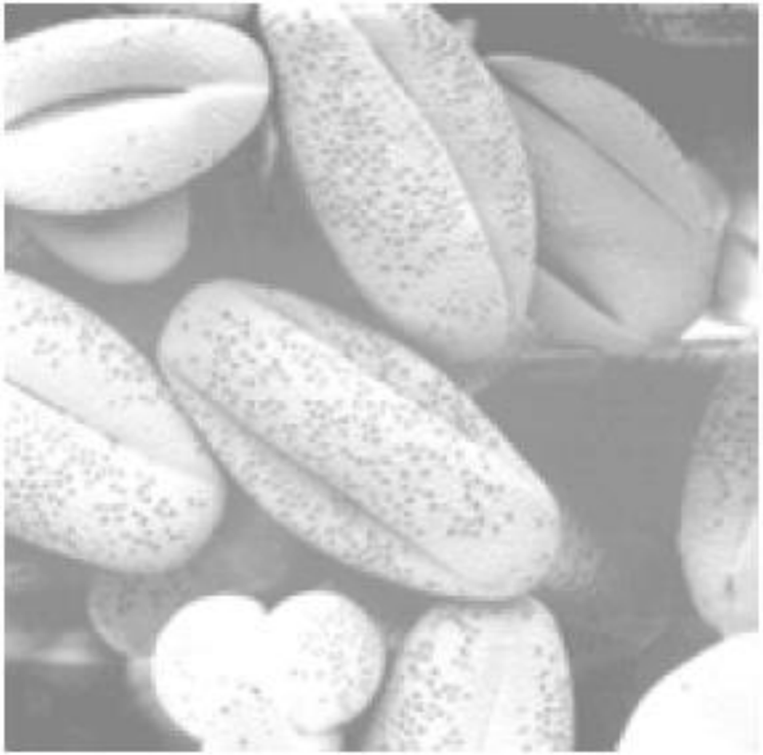
- The histogram of a dark image will be clustered towards the lower gray levels whereas; the histogram of bright images will be clustered towards higher gray levels.
- For a low-contrast image, the histogram will not be spread equally that is the histogram will be narrow. For a high-contrast image, the histogram will have an equal spread in the gray level.

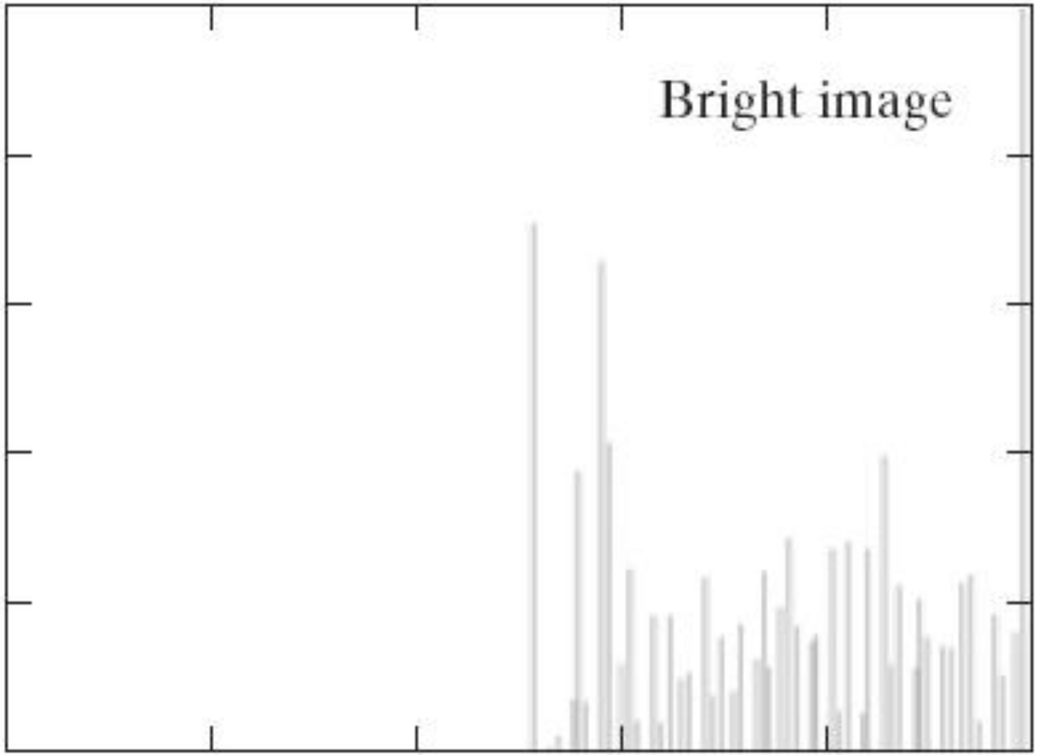
Dark Image



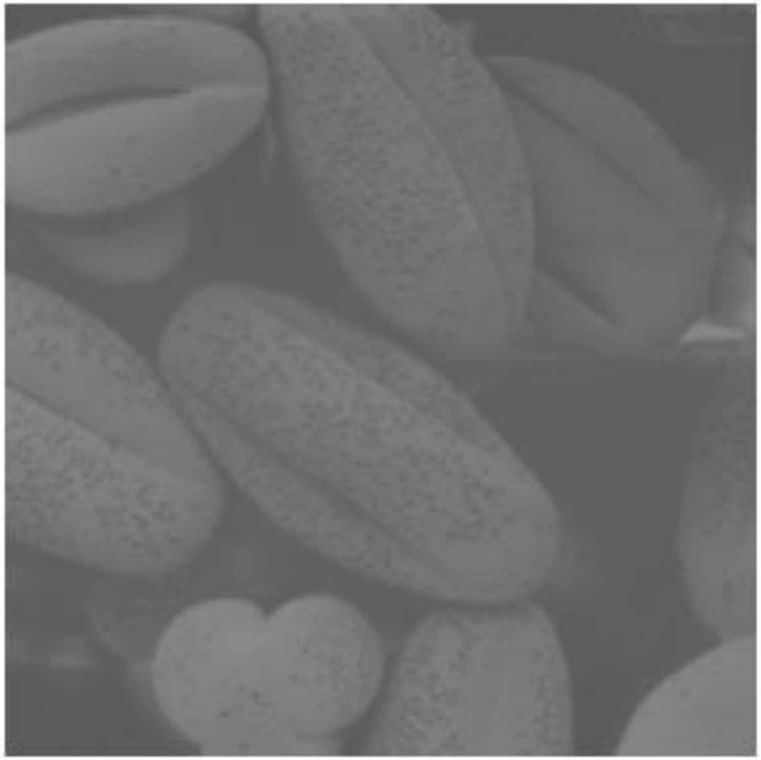


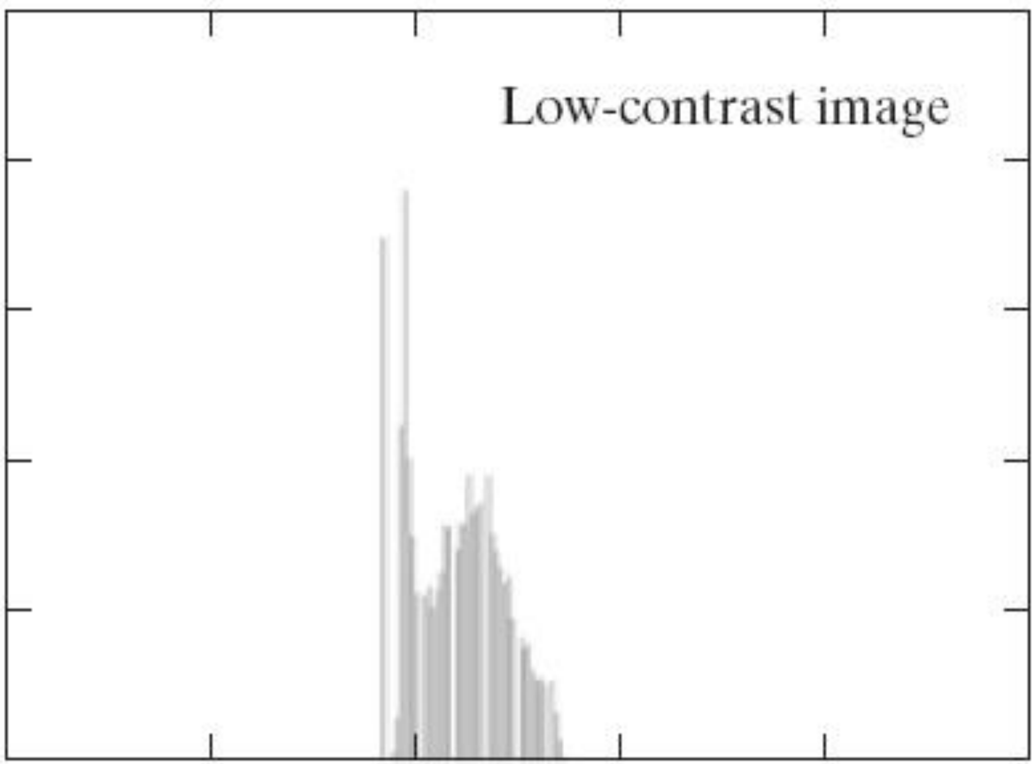
Bright Image



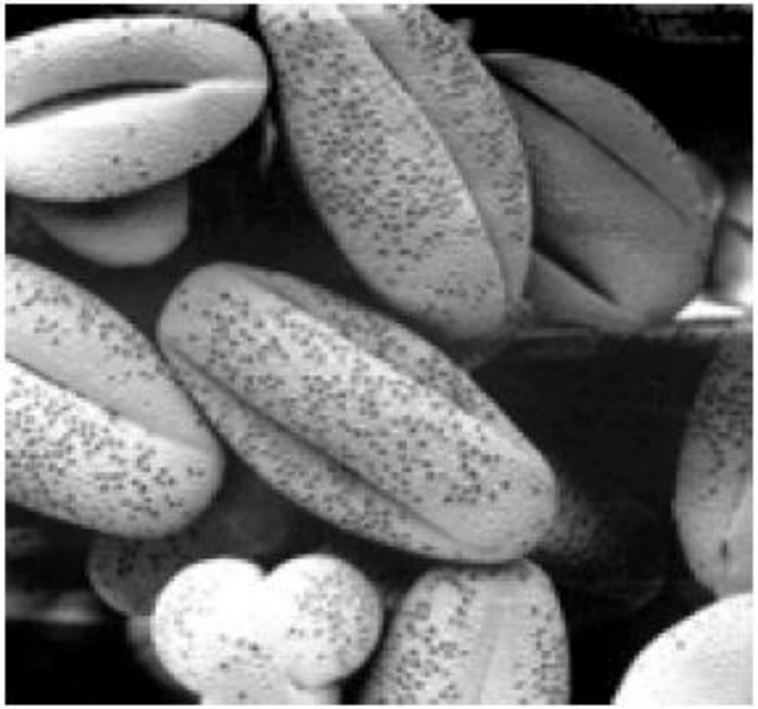


Low Contrast Image





High Contrast Image



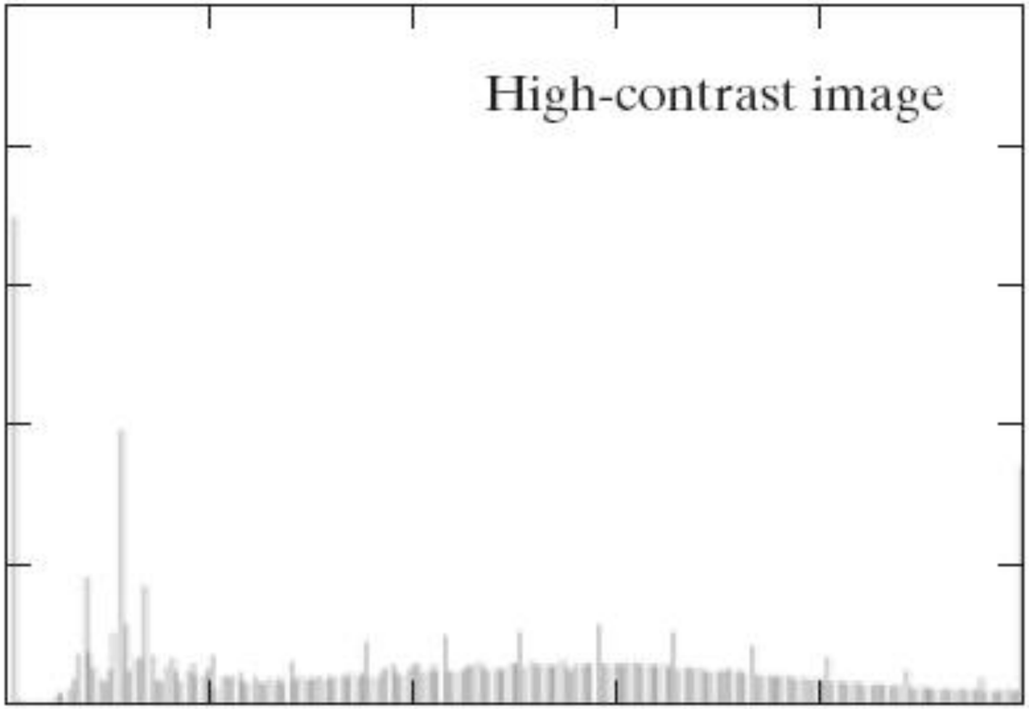


Image Histograms

❖ Histograms clustered at the low end correspond to **dark** images.

❖ Histograms clustered at the high end correspond to **bright** images.

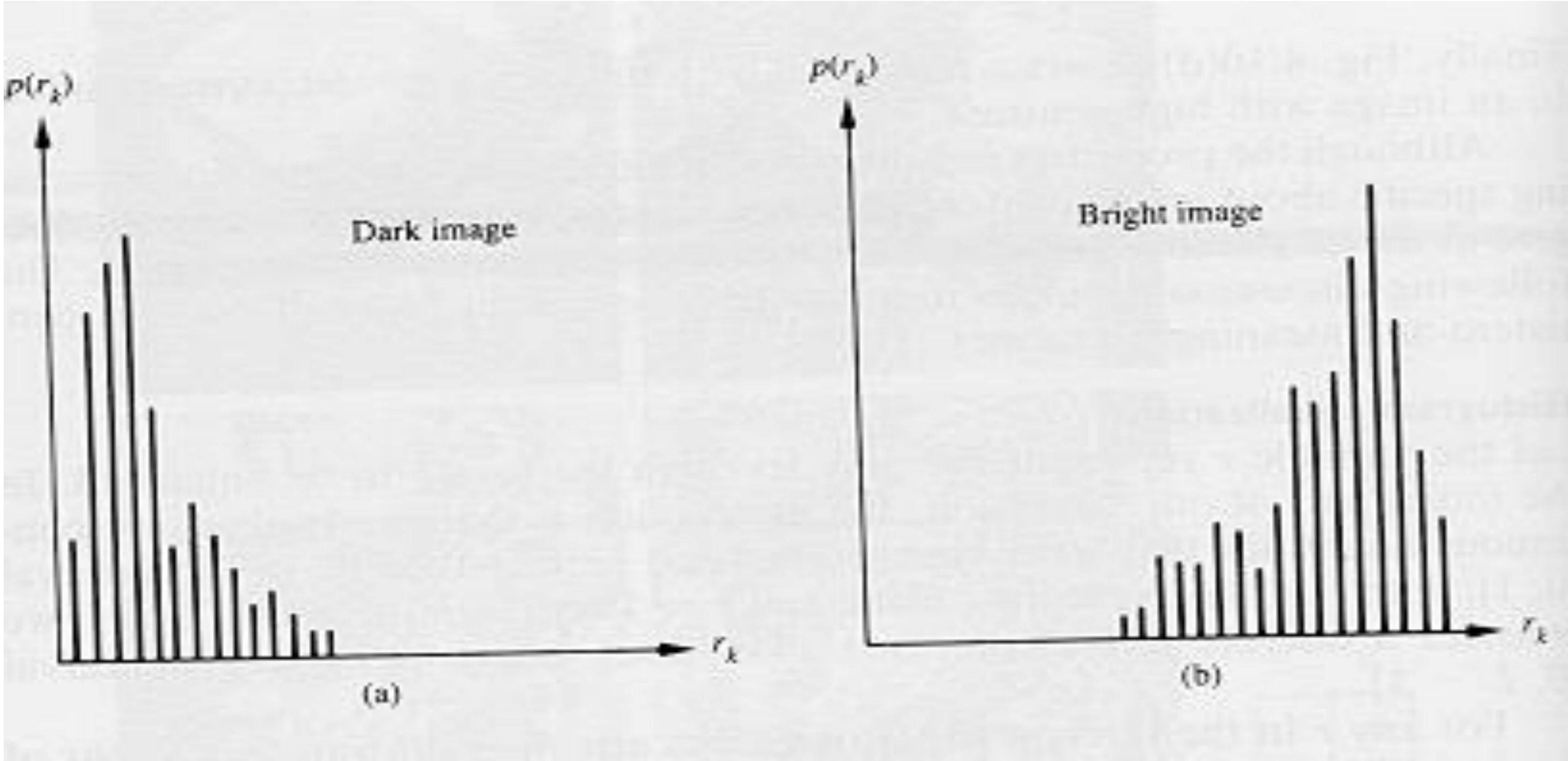


Image Histograms

- Histograms with small spread correspond to **low contrast** images (i.e., mostly dark, mostly bright, or mostly gray).
- Histograms with wide spread correspond to **high contrast**

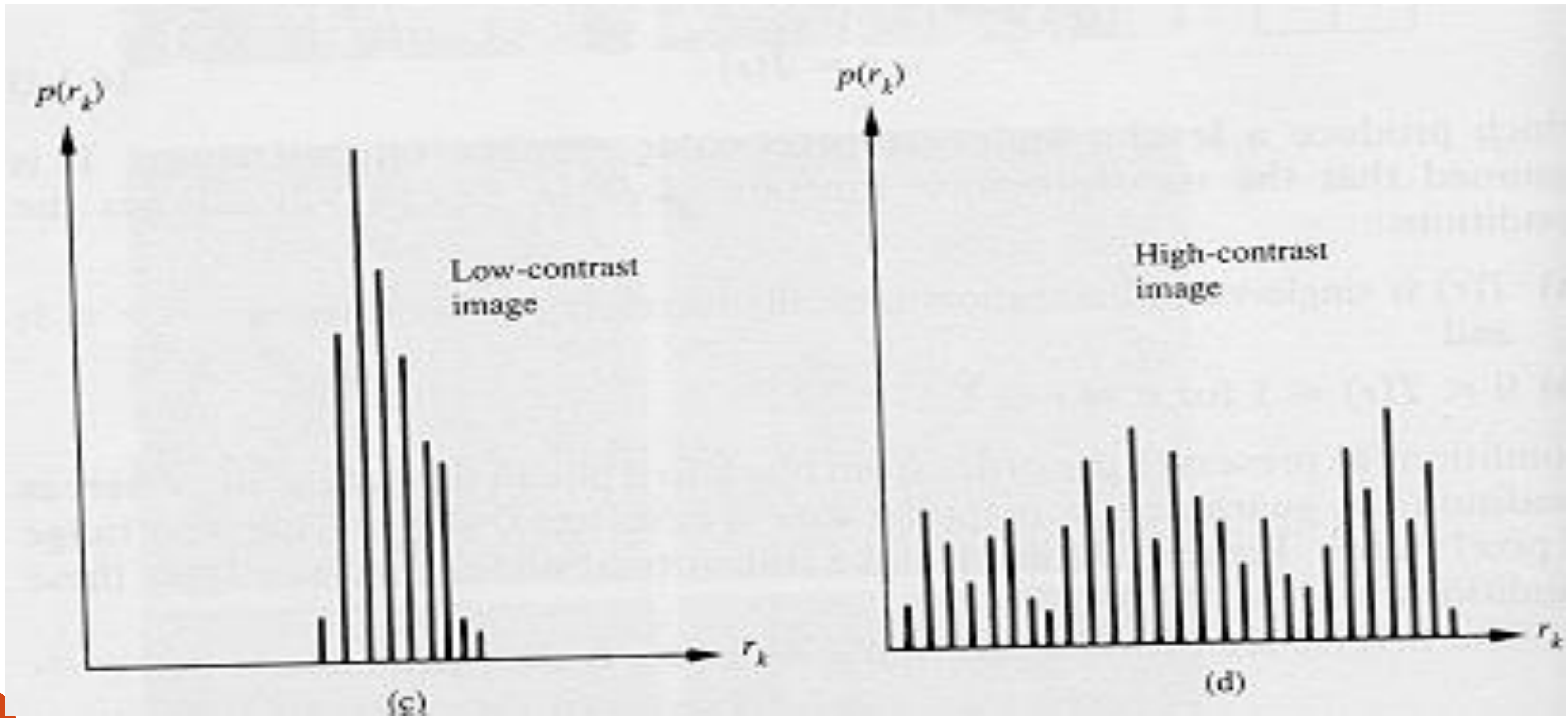
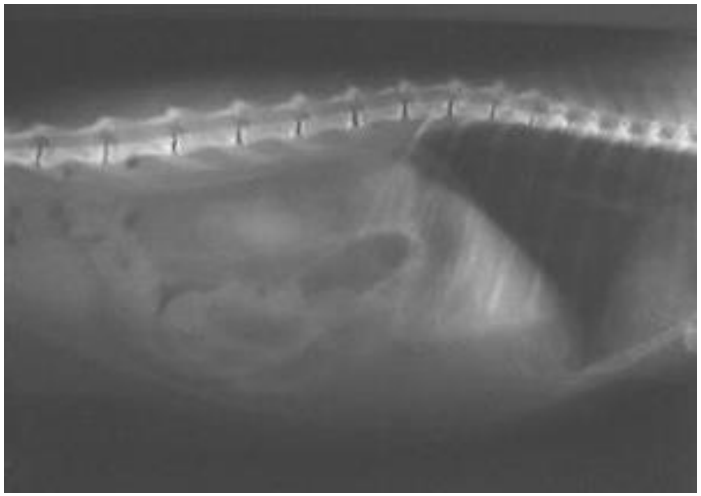
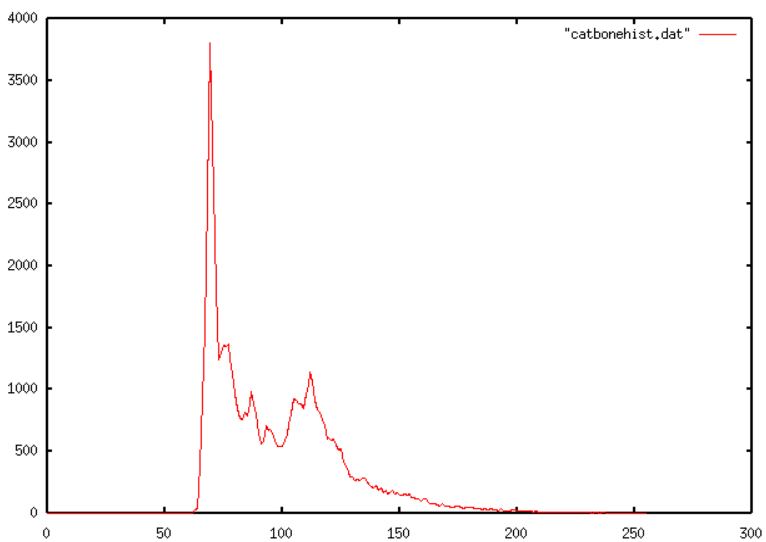


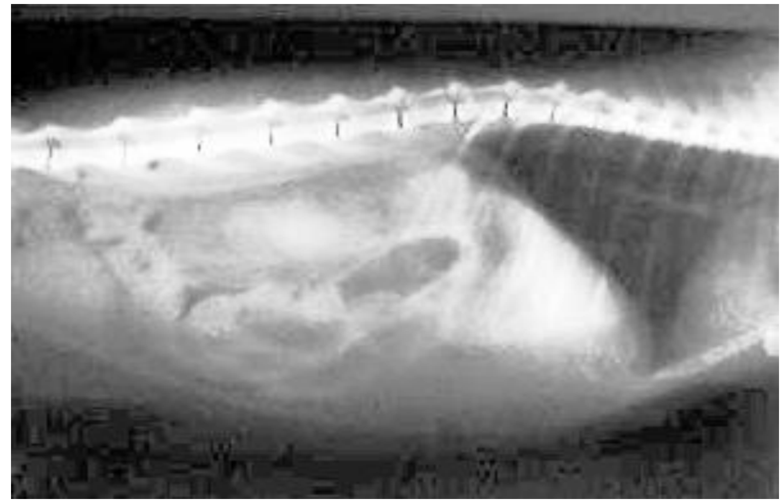
Image Histograms

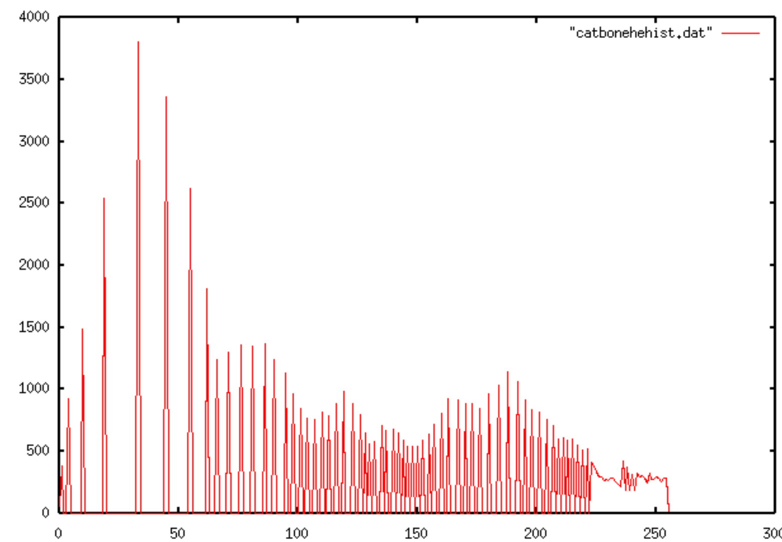
Low contrast





High contrast





Histogram Equalization.

- The histogram technique that is used to enhance the brightness and contrast of an image is histogram equalization.
- The goal of histogram equalization is to distribute the gray levels within an image so that every gray level is equally likely to occur.
- In other words, histogram equalization takes an image's histogram and produces a new image with a histogram that is uniformly distributed.
- Histogram equalization will increase the brightness and contrast of a dark and low contrast image, making features observable that were not visible in the original image.

Histogram Equalization.

- Since histogram equalization distributes an image's gray levels uniformly about the range of gray levels.
- All images will have approximately the same brightness and contrast.

Procedure to perform Histogram Equalisation

- Histogram Equalisation is done by performing the following steps.

I. Find the running sum of the histogram values.
II. Normalise the values from step(1) by dividing by the total number of pixels.
III. Multiply the values from step(2) by the maximum gray level value and round.
IV. Map the gray level values to the results from step(3) using a one to one correspondence.

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

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

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

Step 1 Compute the running sum of histogram values.

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

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

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

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

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

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

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

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

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

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

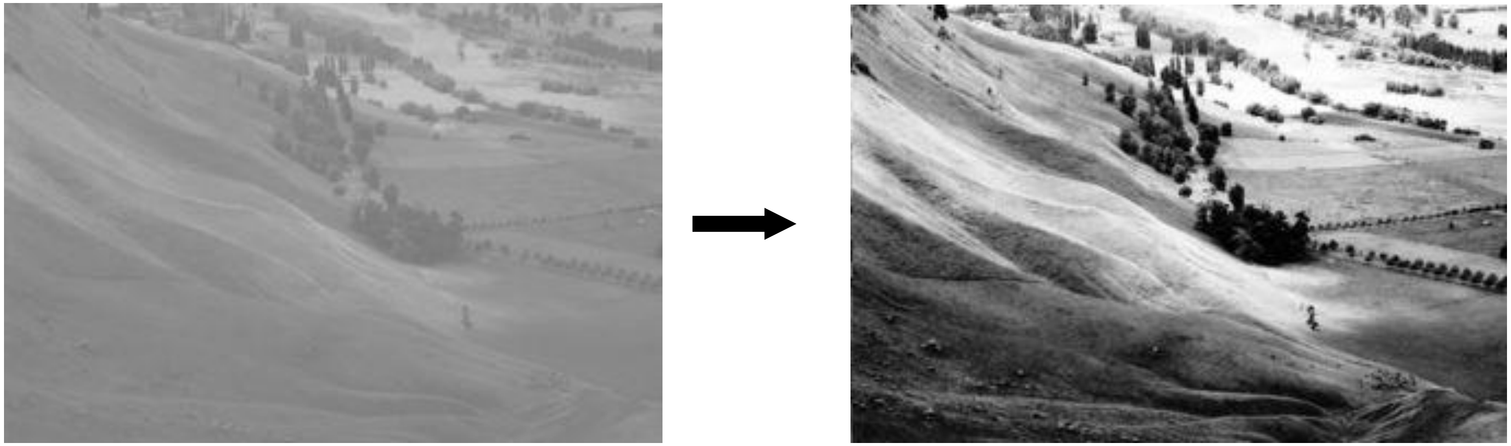
The original image and the histogram equalised image are shown side by side.

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

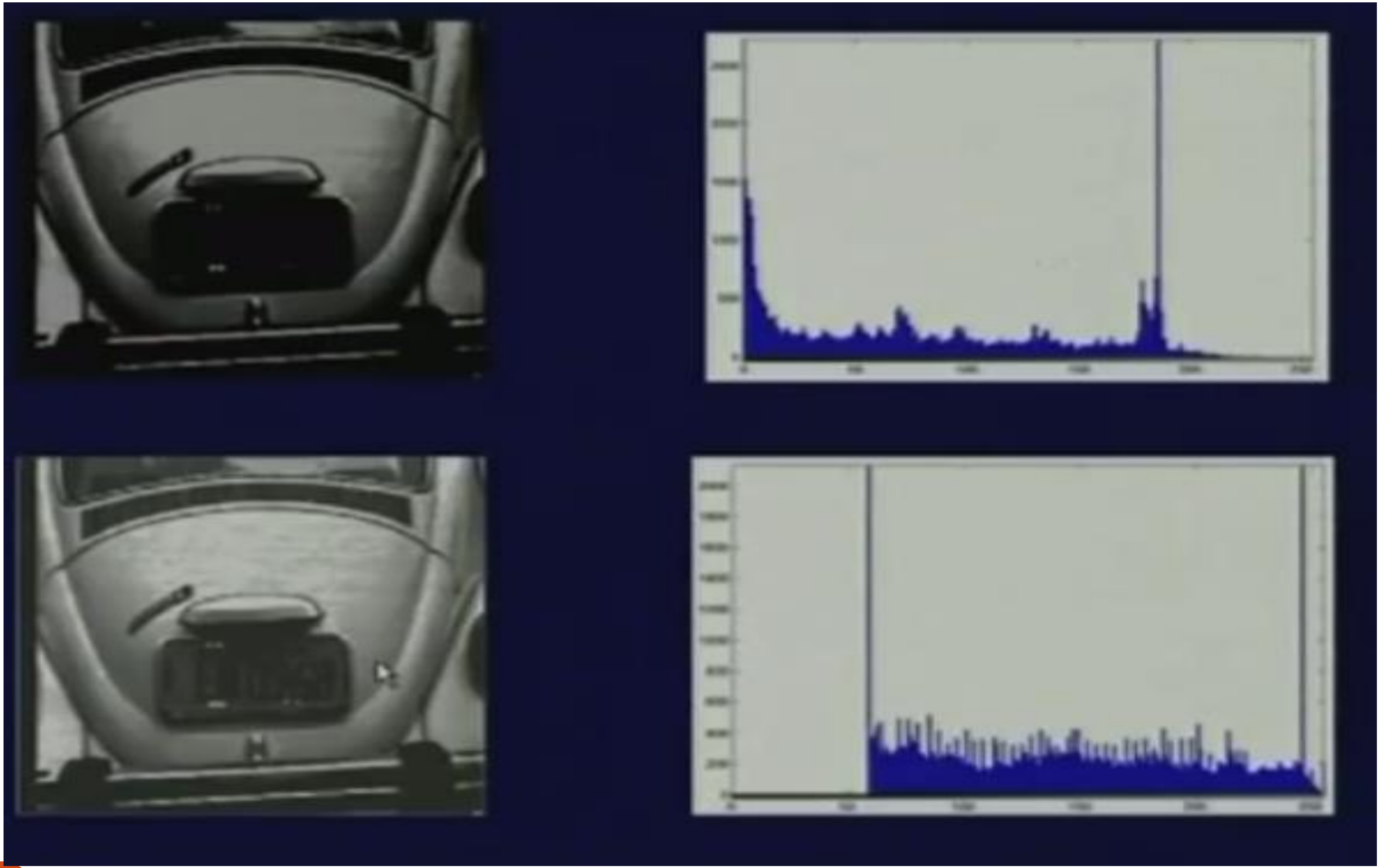
Original image **Histogram equalised image**

Histogram Equalization

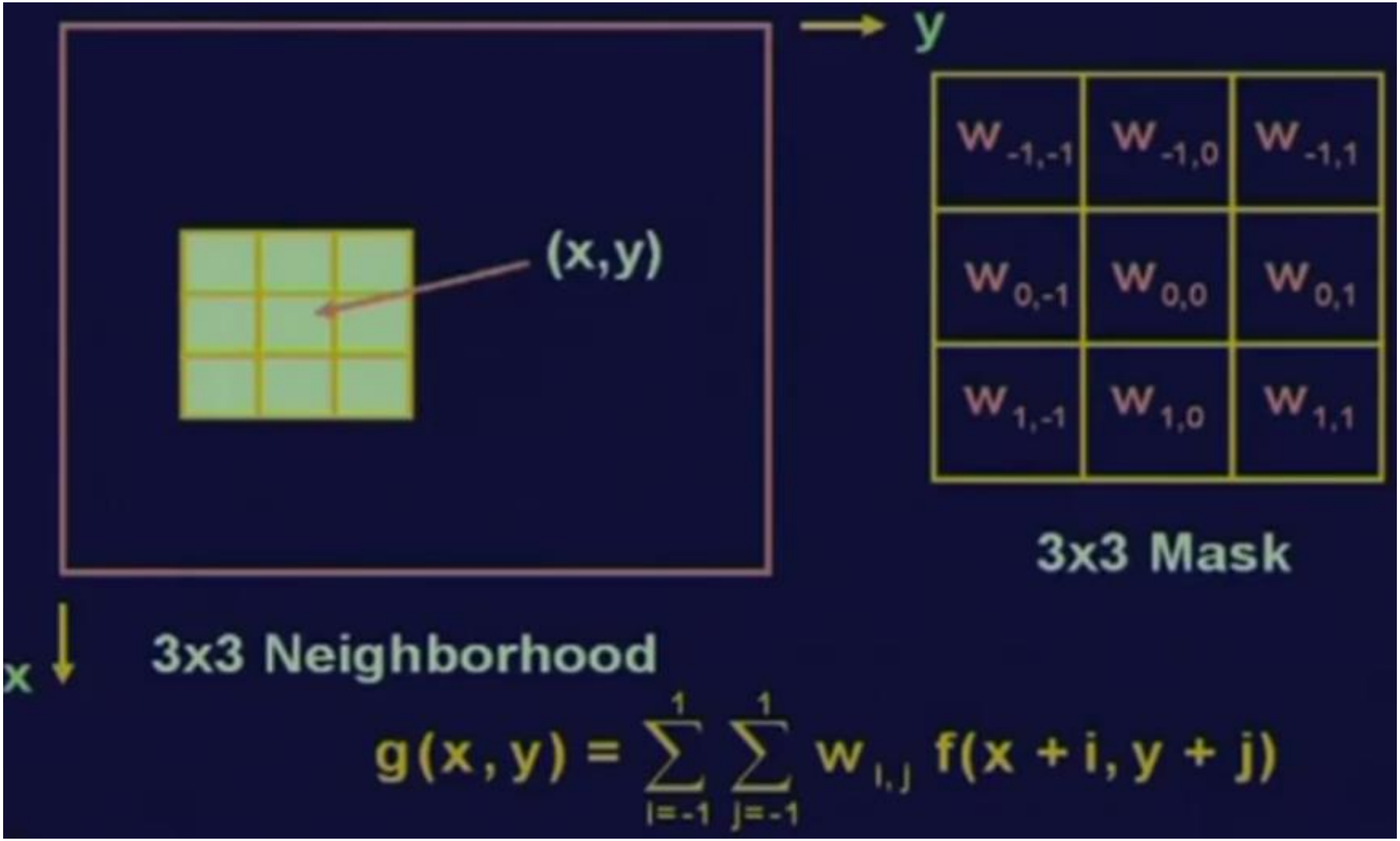
- A fully automatic gray-level stretching technique.



Histogram Equalization



Mask Processing



Mask Processing

- In Mask processing we consider mask 3x3, 5x5,7x7 , 8x8 so on.
- Intensity value g(x,y) in process image depend on mask values Wi,j
- Mask Processing is used for Image Sharpening operation, Image Enhancement operation and Edge detection operation.

Thanks