# Module I

# Introduction to Image Processing Systems:

# 1

# DIGITAL IMAGE PROCESSING

**Unit Structure**

## 1.0 OBJECTIVES

After going through this unit, you will be able to:

- Gain the knowledge about evolution of digital image processing
- Analyse the limits of digital images
- Derive the representation and relationship of pixels
- Describe the functioning of digital image processing system
- Specify the color models of image processing such as RGB, CMY and Hue

## 1.1 INTRODUCTION

Digital Images plays main role in the day-to-day life. The visual effect plays major role than any other media. When we see an image without saying, without explaining anything we understand the concept.

**Evolution of Digital Images:**

The digital images started its role from newspapers. The pictures that are sent through submarine cable between London to New York are the first journey of digital Images.

**1921**

Bartlane cable picture transmission system used specialized printing equipment coded pictures and then reproduced on telegraph printer fitted with typefaces simulating a halftone pattern. This technology reduced the time required to transmit a picture across Atlantic to less than 3 hours.

Level of coding images was 5. Figure 1.1 shows the picture transmitted in this way.

**<u>1922</u>**

Visual quality is improved through selection of printing procedures and distribution of intensity levels. A technique based on photographic reproduction made from tapes perforated at telegraph receiving terminal.

Level of coding images was 5. Figure 1.2 shows the picture transmitted in this way.

**<u>1929</u>**

The intensity level was increased to 15. Figure 1.3 shows the picture transmitted in this way.

**<u>1964</u>**

The digital image used through digital computer and its advanced techniques lead to Digital image processing. The Ranger 7 spacecraft of U.S. took the first image of moon, shown in Figure 1.4. The enhanced methods from the lessons learned from this imaging served as the basis for Surveyor missions to moon, Mariner series missions to Mars and Appolo manned flights to the moon and others.

**Figure 1.1 :**

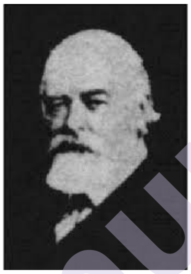


**Figure 1.2 :**

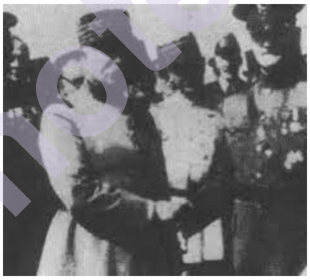


**Figure 1.1 :**

**Figure 1.4 :**



**<u>1970</u>**

In parallel to space applications, the medical imaging, remote earth resources and astronomy the digital image processing was applied. Ex. CAT- Computerized Axial Tomography and X-rays uses DIP.

**<u>1992</u>**

Berners-Lee uploaded the first image to the internet, in 1992. It was of Les HorriblesCernettes, a parody pop band founded by CERN employees.

**<u>1997</u>**

Fractals: Computer generated images are introduced based on the iterative reproduction of a basic pattern according to some mathematical rules.

## 1.2 AN OVERVIEW

### 1.2.1. What is an Image?

Visual representation of an object is called as Image. An image is a two-dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene.

Fig. 1.5. Sample Image[1]

(1 https://www.designyourway.net/diverse/amazingworld/28899053723.jpg)

### 1.2.2 What is a digital image?

Digital image is composed of a finite number of elements having a particular location and value. These elements are called picture elements, image elements, pels and pixels

A real image can be represented as a two dimensional continuous light intensity function g(x,y) where x and y denote the spatial coordinates and the value of g is proportional to the brightness (or gray level) of the image at that point.

### 1.2.3 Types of Image

Generally the images can be classified into two types. They are

i) Analog Image

ii) Digital Image

***i)Analog Image***

The image which is having continuously varying physical quantity in the spatial data such as x, y of the particular axis is known as Analog Image. Analog image can be mathematically represented as a continuous range of values representing position and intensity. The image produced on the screen of a CRT monitor, Television and medical images are analog images.

***ii) Digital Image***

A digital image is composed of picture elements called pixels with discrete data. Pixels are the smallest sample of an image. A pixel represents the brightness at one point. The common formats of digital images are TIFF, GIF, JPEG, PNG, and Post-Script.

***Advantages of Digital Images***

i) The processing of images is faster and cost-effective.

ii) Digital images can be effectively stored and efficiently transmitted from one place to another.

iii) Immediate output display to see the image.

iv) Copying a digital image is easy. The quality of the digital image will not be degraded even if it is copied for several times.

v) The reproduction of the image is both faster and cheaper.

vi) Digital technology supports various image manipulations.

***Drawbacks of Digital Images***

i) Misuse of images has become easier.

ii) During enlarging the image, the quality of the image will be compromised.

iii) Large volume of memory is required to store and process the images.

iv) Fast processors required to process digital image processing algorithms.

**1.2.4. Digital Image Processing (DIP)**

Processing the images using digital computers is termed as *Digital Image Processing*.

Digital image processing concepts are allied in the fields of defence, medical diagnosis, astronomy, archaeology, industry, law enforcement, forensics, remote sensing etc.

***Flexibility and Adaptability***

Modification in hardware components is not required in order to reprogram digital computers to solve different tasks. This feature makes digital computers an ideal device for processing image signals adaptively.

***Data Storage and Transmission***

The digital data can be effectively stored since the development of different image compression algorithm is in progress. The digital data can be easily transmitted from one place to another and from one device to another using the computer and its technologies.

Different image processing techniques include image enhancement, image restoration, image fusion and image watermarking for its effective applications.

## 1.3 IMAGE REPRESENTATION

- Represented as M □ N matrix.
- Each element in the matrix is a number that represents sampled intensity.
- M □N gives resolution by pixel.

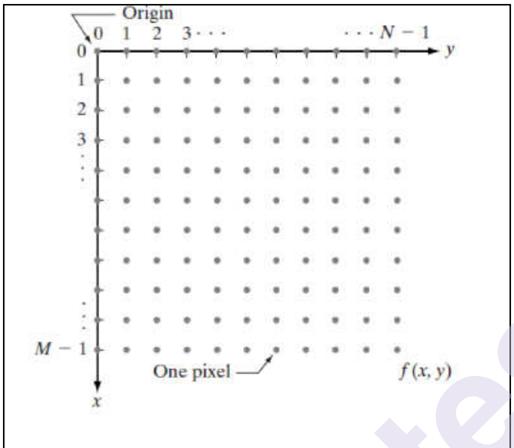



Figure 1.6. Coordinate convention used to represent digital images.

Digital image is a finite collection of discrete data samples (pixels) of any visible object. The pixels represent a two or higher dimensional "view" of the object, each pixel having its own discrete value in a finite range. The pixel values may represent the amount of visible light, infra-red light, absorption of x-rays, electrons, or any other measurable value such as ultrasound wave impulses.

The result of sampling and quantization is matrix of real numbers. Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N Columns. The values of the coordinates (x,y) now become discrete quantities thus the value of thecoordinates at origin become (x,y) = (0,0). The next Coordinates value along the first signify the image along the first row.

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1)\ ) & f(1,2) & \cdots & f(1,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f(M-1,0) & f(M-1,1)) & f(M-1,2) & \cdots & f(M-1,N-1) \\ f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1)\ ) & f(1,2) & \cdots & f(1,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f(M-1,0) & f(M-1,1)) & f(M-1,2) & \cdots & f(M-1,N-1) \end{matrix}$$

Fig 1.7 Matrix representation format of a digital image

The right side of this equation is by definition a digital image. Each element ofthis matrix array is called an *image element*, *picture element*, *pixel*, or *pel*.

Or the same can be represented as

$$A= \begin{bmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ \vdots & \ddots & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{bmatrix} \begin{bmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ \vdots & \ddots & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

## 1.4 BASIC RELATIONSHIP BETWEEN PIXELS

There are several important relationships between pixels in a digital image.

### 1.4.1 Neighbors of a Pixel

A pixel $p$ at coordinates (x,y) has four horizontal and vertical neighbours whose coordinates are givenby:

This set of pixels, called the 4-neighbors of p, is denoted by $N_4(p)$. Each pixel is one unit distance from (x,y) and some of the neighbors of $p$ lie outside the digital image if (x,y) is on the border of the image. The four diagonal neighbors of $p$ have coordinates and are denoted by $N_D(p)$.

.These points, together with the 4-neighbors, are called the 8-neighbors of $p$, denoted by $N_8(p)$.

| | | |
|---|---|---|
| | (x,y-1) | |
| (x-1, y) | p(x,y) | (x+1,y) |
| | (x, y+1) | |

### 1.4.2 Adjacency, Connectivity, Regions and Boundaries

- To define adjacency the set of grey–level values V is considered.
- In a binary image, the adjacency of pixels with value 1 is referred as V={1}.
- In a grey-scale image, the idea is the same, but Vtypically contains more elements for example, V= {100, 101,…,150} that is subset of any 256 values from 0-255

***Types of Adjacency:***

(i) 4- Adjacency – two pixels *p* and *q* with value from V are 4 – adjacency if A is in the set $N_4(p)$

(ii) 8- Adjacency – two pixels *p* and *q* with value from V are 8 –adjacency if A is in the set $N_8(p)$

(iii) M-adjacency –two pixel *p* and *q* with value from V are m – adjacency if

a) Q is in $N_4$(p) or

b) Q is in $N_D$(q) and theSet $N_4(p) \cap N_4(q)$ has no pixel whose values are fromV.

Mixed adjacency is a modification of 8-adjacency. It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Fig.1.8 Arrangement of pixels

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Fig.1.9 pixels that are 8-adjacent (dashed lines) to the center pixel

$$\begin{matrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Fig.1.10 m-adjacency

***Digital Path:***

A digital path from pixel *p*(x,y) to pixel *q*(s,t) is a sequence of distinct pixels with coordinates $(x_0,y_0)$, $(x_1,y_1)$, …, $(x_n, y_n)$ where $(x_0,y_0) = (x,y)$ and $(x_n, y_n) = (s,t)$ and pixels $(x_i, y_i)$ and $(x_i-1, y_i-1)$ are adjacent for $1 \le i \le n$, n is the length of the path.

If $(x_0,y_0) = (x_n, y_n)$, the path is closed.

Based on the type of adjacency paths are specified as 4, 8 or m-paths.

In figure 1.9 the paths between the top right and bottom right pixels are 8-paths. And the path between the same 2 pixels in figure 1.10 is m-path

***Connectivity:***

Let S represent a subset of pixels in an image, two pixels *p* and *q* are said to be connected in S if there exists a path between them consisting entirely of pixels in S.

For any pixel *p* in S, the set of pixels that are connected to it in S is called a connected component of S. If it only has one connected component, then set S is called a *connected set.*

***Region and Boundary:***

*Region:* Let R be a subset of pixels in an image, R is a region of the image if R is a connected set. Any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the region boundary.

*Boundary:* The boundary of a region R is the set of pixels in the region that have one or more neighbors that are not in R.

If R is an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns in the image. There are no neighbors beyond the pixels' borders.

**.1.4.3. Distance Measures**

For pixel *p,q* and z with coordinate (x.y) ,(s,t) and (v,w) respectively D is a distance function or metric if

D [p.q] $\geq$ 0 {D[p.q] = 0 iff p=q

D [p.q] = D [p.q] and

D [p.q] $\geq$ 0 {D[p.q]+D(q,z)

The Euclidean Distance between p and q is defined as:

$$De(p,q) = [(x-s)^2 + (y-t)^2\ ]^{1/2}$$
$$De(p,q) = [(x-s)^2 + (y-t)^2\ ]^{1/2}$$

Pixels having a distance less than or equal to some value r from (x,y) are the points contained in a disk of radius " r" centered at (x,y)

The $D_4$distance (also called city-block distance) between p and q is definedas:

$D_4(p,q) = | x – s | + | y – t |$

Pixels having a $D_4$ distance from (x,y), less than or equal to some value r form a Diamond centered at (x,y)

**Example:**

The pixels with distance $D_4 \leq 2$ from (x,y) form the following contours of constant distance.

The pixels with $D_4$= 1 are the 4-neighbors of (x,y)

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

The $D_8$distance (also called chessboard distance) between p and q is defined as:

$D_8(p,q) = \max(| x - s |,| y - t |)$

Pixels having a $D_8$ distance from (x,y), less than or equal to some value r form a square Centered at (x,y).

| | | | | |
|---|---|---|---|---|
| 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 |

**Example:**

$D_8$distance $\leq$ 2 from (x,y) form the following contours of constant distance.

**$D_m$Distance:**

$D_m$ is the shortest m-path between the points.In this case, thedistance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

Example:

| | | |
|---|---|---|
| | $P_3$ | $P_4$ |
| $P_1$ | $P_2$ | |
| p | | |

Consider the following arrangement of pixels and assume that p, p2, and p4 have value 1 and that p1 and p3 can have can have a value of 0 or 1

Consider the adjacency of pixels values; V ={1}.Compute the $D_m$ between points p and $p_4$

There are 4 cases:

p $\quad$ $p_2$ $\quad$ $p_4$

*Case1:* If $p_1$ =0 and $p_3$ = 0

Length of the shortest m-path (the $D_m$ distance) is 2;

*Case2:* If $p_1$ =1 and $p_3$ = 0

$p_1$ and p will no longer be adjacent then, the length of the shortest path will be 3

p $\quad$ $p_1$ $\quad$ $p_2$ $\quad$ $p_4$

*Case3:* If $p_1$ =0 and $p_3$ = 1

p $\quad$ $p_2$ $\quad$ $p_3$ $\quad$ $p_4$

The shortest –m-path will be 3 ;

*Case4:* If $p_1$ =1 and $p_3$ = 1

p $\quad$ $p_1$ $\quad$ $p_2$ $\quad$ $p_3$ $\quad$ $p_4$

The shortest –m-path will be 4 ;

### 1.4.4 Image operations on a Pixel Basis

For doing arithmetic and logic operations between the images, the corresponding pixels in the images are involved in those operations.

If any image is divided by another then the division is carried out between the corresponding pixels in the two images.

Let f and g are two images

Applying the division operation, h=f/g

First element of image ‘h’ is the resultant of first pixel of image ‘f’ divided by image ‘g’

## 1.5 ELEMENTS OF DIGITAL IMAGE PROCESSING SYSTEMS:

The basic elements of digital image processing systems are

i) Image Acquisition devices

ii) Image storage devices

iii) Image processing elements

iv) Image display devices

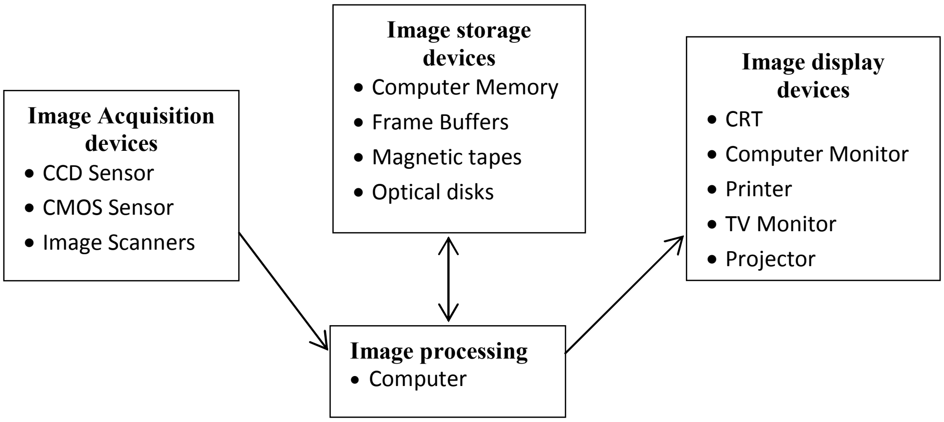



Fig. 1.11 Elements of DIP system

## i) Image Acquisition devices

The term image acquisition refers to the process of capturing real-world images and storing them into a computer. Conventional silver-based photographs in the form of negatives, transparencies or prints can be scanned using a variety of scanning devices. Digital cameras which capture images directly in digital form are more popular nowadays. Films are not used in digital cameras. Instead, they use a charge-coupled device or CMOS device as the image sensor that converts light into electrical charges. An image sensor is a 2D array of light-sensitive elements that convert photons to electrons. Most of the digital cameras use either a CCD or a CMOS image sensor.

Solid-state image sensor consists of

a) Discrete photo-sensing elements b) charge-transport mechanism c) an output circuit.

- ❖ The photo sensitive sites convert the incoming photons into electrical charges and integrate these charges into a charge packet.
- ❖ The charge packet is then transferred through the transport mechanism to the output circuit where it is converted into a measurable voltage.
- ❖ The types of photo-sensing elements used in solid state imagers include photodiodes, MOS capacitors, Schottky-barrier diodes and photoconductive layers.
- ❖ The output circuit typically consists of a floating diffusion and source-follower amplifier.
- ❖ In practical applications, image sensors are configured in a one-dimensional (linear devices) or a two-dimensional manner.

***ii) Image storage devices***

If the image is not compressed the enormous volume of storage is required

There are three categories of storage devices. They are :

a) Short term storage b) Online storage c) Archival Storage

*Short term storage* : Used at the time of processing, Example: computer memory, frame buffers. Frame buffers stores more than one image and can be accessed rapidly at video rates. Image zoom, scrolling and pan shifts are done through frame buffers.

*Online storage*: It is used while accessing the data often. It encourages the fast recall, Example; magnetic disk or optical media.

*Archival storage:* It is characterized by frequent access, example: magnetic tapes and optical disks. It requires large amount of storage space and the stored data is accessed infrequently.

***iii) Image processing elements***

Computer and its related devices are the image processing elements for various applications.

*iv) Image display devices*

Image displays are color TV monitors. These monitors are driven by the output of image and graphics display cards which are a part of the computer system.

## 1.6 ELEMENTS OF VISUAL PERCEPTION

### 1.6.1 Structure of Human Eye

*Characteristics of Eye*

- Nearly spherical
- Approximately 20 mm in diameter
- Three membranes

i) Cornea and Sclera

ii) Choroid

iii) Retina

***i) Cornea; Sclera***

The cornea is a tough, transparent tissue that covers the anterior , front surface of eye. The sclera is an opaque membrane that is continuous with the cornea and encloses the remaining portion of the eye.

## ii) Choroid

It is located directly below the sclera. It contains network of blood vessels which provides nutrition to the eye. The outer cover of the choroid is heavily pigmented to reduce amount of extraneous light entering the eye. Also contains the iris diaphragm and ciliary body

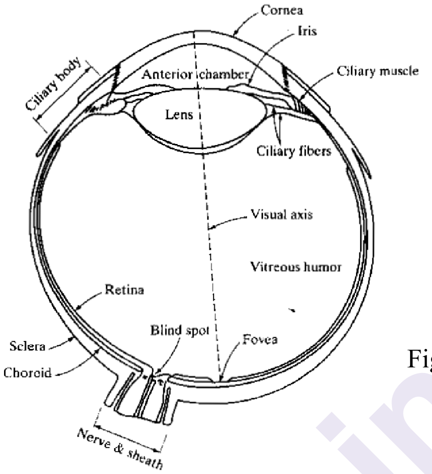



Fig. 1.12 Structure of human Eye

## Iris diaphragm

It contracts and expands to control the amount of light entering into the eye. The central opening of the iris which appears black is known as ***pupil*** whose diameter varies from 2mm to 8mm.

## Lens

It is made up of many layers of fibrous cells. It is suspended and is attached to the ciliary body. It contains 60% to 70% water and 6% fat and more protein. The lens is colored by a slightly yellow pigmentation. This coloring increases with age, which leads to clouding of lens. Excessive clouding of lens happens in extreme cases which are known as **cataracts**. This leads to poor color discrimination and loss of clear vision.

The lens absorbs approximately 8% of the visible light spectrum, with relatively higher absorption at shorter wavelengths. Both infrared and ultraviolet light are absorbed appreciably by proteinswithin the lens structure and, in excessive amounts, can damage the eye.

## iii) Retina

It is the inner most membrane, objects are imaged on the surface. The central portion of retina is called the *fovea*. Two types of receptors in retina are Rods and Cones

Rods are long small receptors and Cones are short thicker in structure.The rods and cones are not distributed evenly around the retina.

### ***Cones***

Cones are highly sensitive to color and are located in the *fovea*. There are 6 to 7 million cones. Each cone is connected with its own nerve end. Therefore humans can resolve fine details with the use of cones. Cones respond to higher levels of illumination; their response is called ***photopic vision or bright light vision***

### ***Rods***

Rods are more sensitive to low illumination than cones. There are about 75 to 159 million rods. Many numbers of rods are connected to a common, single nerve. Thus the amount of detail recognizable is less. Therefore rods provide only a general overall picture of the field of view. Due to stimulation of rods the objects that appear color in daylight will appear colorless in moon light. This phenomenon is called ***scotopic vision or dim light vision.***

The area where there is absence of receptors is called the blind spot

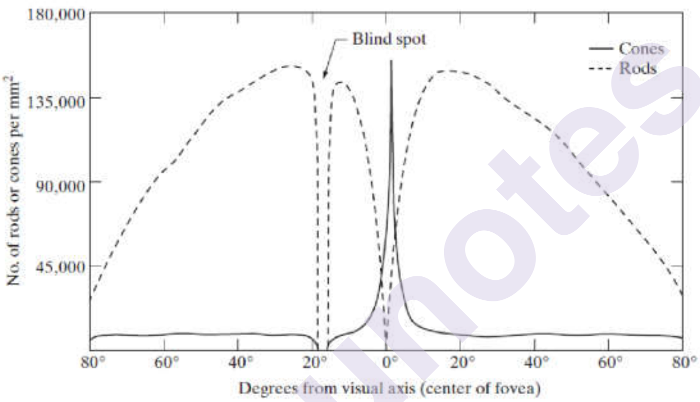



**Fig 1.13 Rods and Cones in Retina**

Receptor density measured in degrees from the fovea (the angle formed between the visual axis and a line extending from the center of the lens to the retina

### 1.6.2 Image Formation in the Eye

The lens of eye is flexible, whereas the optical lens is not.

The radius of curvature of the anterior surface of the lens is greater than the radius of its posterior surface.

The tension in the fibers of the ciliary body controls the shape of the lens

To focus distant object greater than 3m the lens is made flattened by the controlling muscles and it will have lowest refractive index

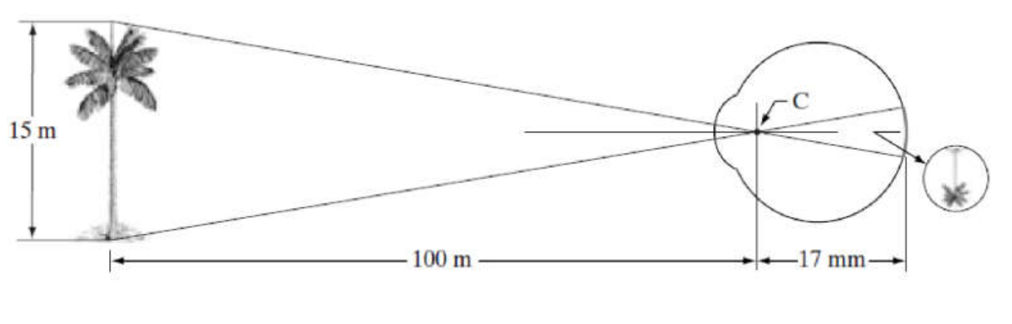



Fig. 1.14 Graphical representation of the eye Point $C$ is the optical center of the lens

To focus nearer objects the muscles allow the lens to become thicker,and strongest refractive index.

The distance between the centre of the lens and the retina is called focal length.

It ranges from 14mm to 17mm as the refractive power decreases from maximum to minimum.

### 1.6.3 Brightness

The following terms are used to define color light:

i)Brightness or Luminance: This is the amount of light received by the eye regardless of color.

ii) Hue: This is the predominant spectral color in the light.

iii)Saturation: This indicates the spectral purity of the color in the light

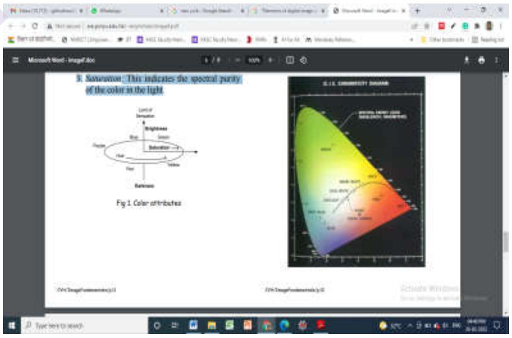


**Fig. 1.15 Color attributes**

The range of light intensity levels to which the human visual system can adapt is enormous from scotopic threshold to the glare limit. Subjective brightness is a logarithmic function of the light intensity incident on the eye.

*Brightness adaptation* :The human visual system has the ability to operate over a wide range of illumination levels. Dilation and contraction of the iris of the eye can account for a change of only 16 times in the light intensity falling on the retina. The process which allows great extension of this range by changes in the sensitivity of the retina is called brightness adaptation .

### 1.6.4 Contrast

The response of the eye to changes in the intensity of illumination is non-linear

This does not hold at very low or very high intensities and it is dependent on the intensity of the surround.

Perceived brightness and intensity

Perceived brightness is not a function of intensity. This can be explained by Simultaneous contrast and Mach band effect

*Simultaneous contrast*

The small squares in each image are the same intensity.

Because the different background intensities, the small squares do not appear equally bright.

Perceiving the two squares on different backgrounds as different, even though they are in fact identical, is called the simultaneous contrast effect.

Psychophysically, we say this effect is caused by the difference in the backgrounds.

The term contrast is used to emphasise the difference in luminance of objects. The perceived brightness of a surface depends upon the local background which is illustrated in Fig. 1.16. In Fig. 1.16, the small square on the right-hand side appears brighter when compared to the brightness of the square on the left-handside, even though the gray level of both the squares are the same. This phenomenon is termed ‘simultaneous contrast’. It is to be noted that simultaneous contrast can make the same colours look different.

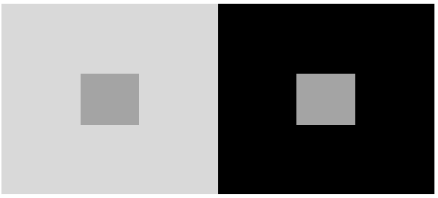
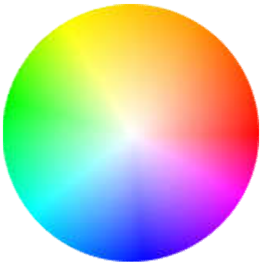


Fig. 1.16 Simultaneous contrast

### 1.6.5 Hue

Hue refers to the dominant color family like Yellow, Orange, Red, Violet, Blue, and Green tertiary colors would also be considered hues. Hue is mixed colors where neither color is dominant.



The pure hues are around the perimeter. The closer to the center of the circle are more desaturated the colors, with white at the center. This Fig 1.17 shows hues, saturation and lightness.



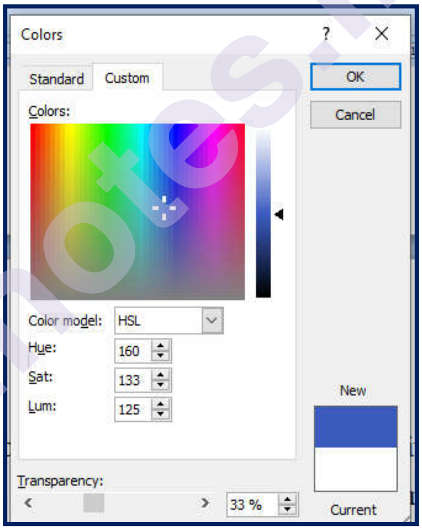



Fig. 1.17 Hue

### 1.6.6 Saturation

Saturation is how "pure" the color is. For example, if its hue is cyan, its saturation would be how purely cyan it is. Less saturated would mean more whitish or grayish. If a color has greater-than-0 values for all three of its red, green and blue primaries then it's somewhat desaturated.

### 1.6.7 Machband effect

The Mechband describes an effect where the human brain subconsciously increases the contrast between two surfaces with different luminance**.** The Mcehband effect is described in Fig 1.18. The intensity is uniform over the bar.

Visual appearance of each strip is darker at its leftside than its right. The special interaction of luminance from an object and its surrounding creates the Mechband effect which shows that brightness is not a monotonic function of luminance.

Mechband is caused by lateral inhibition of receptors in the eye.

Receptors receive the light they draw light-sensitive chemical compound

Receptors directly on the lighter side of the boundary can pull in unused chemicals from the darker side, and produce a stronger response,and the darker side of the boundary, gives a weaker effect..

Luminance within each block is constant

The apparent lightness of each strip vary across its length.

Close to the left edge of the strip it appears lighter than at the centre, and close to the right edge of the strip it appears darker than at the centre.

The visual system is exaggerating the difference in luminance (contrast) at each edge in order top detect it.

It shows that the human visual system tends to undershoot or overshoot around the boundary regions of different intensities.

1.18. Machband Effect

- The intensity is uniform over the width of each bar.
- However, the visual appearance is that each strip is darker at its right side than its left.

## 1.7 SIMPLE IMAGE FORMATION MODEL

An image is denoted by a two dimensional function of the form f{x, y}. The value or amplitude of f at spatial coordinates {x,y} is a positive scalar quantity whose physical meaning is determined by the source of the image. When an image is generated by a physical process, its values are proportional to energy radiated by a physical source. As a consequence, f(x,y) must be non-zero and finite; that is o<f(x,y) <co

The function f(x,y) may be characterized by two components-

i) *Illumination Component:*The amount of the source illumination incident ***i(x,y)*** on the scene being viewed;

*ii) Reflectance components:*The amount of the source illumination r(x,y) reflected back by the objects in the scene.

The functions combine as a product to form f(x,y). The intensity of a monochrome image at any coordinates (x,y) the gray level (l) of the image at that point l= f (x, y.)

$L_{min} \leq l \leq L_{max} L_{min}$is to be positive

$L_{max}$must be finite

$L_{min}=i_{min}r_{min}$

$L_{max} =i_{max}r_{max}$

The interval [$L_{min}$, $L_{max}$] is called gray scale. The interval [0, L-l] where l=0 is considered black and l= L-1 is considered white on the gray scale. All intermediate values are shades of gray of gray varying from black to white.

## 1.8 VIDICON AND DIGITAL CAMERA WORKING PRINCIPLE

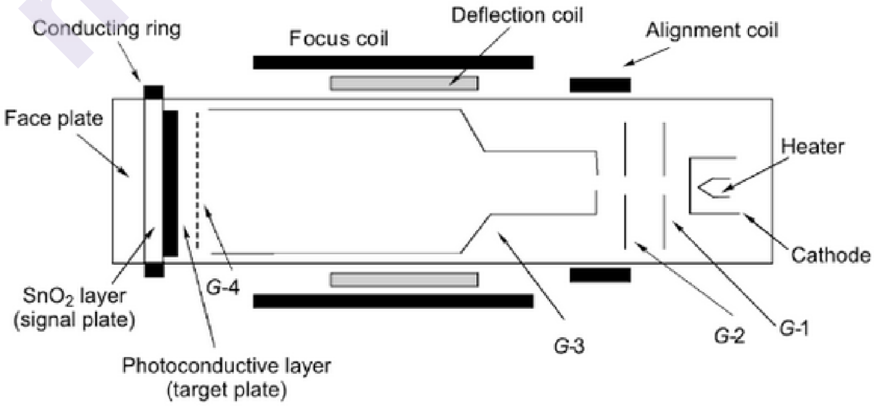
**Vidicon**

The vidicon is a storage-type camera tube in which a charge-density pattern is formed by the imaged scene radiation on a photoconductive surface which is then scanned by a beam of low velocity electrons.

The Vidicon operates on the principle of photo conductivity, where the resistance of the target material shows a marked decrease when exposed to light.

Vidicon is a short tube with a length of 12 to 20 cm and diameter between 1.5 and 4 cm.

Its life is estimated to be between 5000 and 20,000 hours.




The target consists of a thin photo conductive layer of eitherselenium or antimony compounds which behaves like an insulator.

This is deposited on a transparent conducting film, coated on theinner surface of the face plate. This conductive coating is known assignal electrode or plate.

With light focused on it, the photon energy enables more electronsto go to the conduction band and this reduces its resistivity.

Image side of the photolayer, which is in contact with the signalelectrode, is connected to DC supply through the load resistance.

The beam that emerges from the electron gun is focused on surfaceof the photo conductive layer by combined action of uniformmagnetic field of an external coil and electrostatic field of grid No 3.

Grid No. 4 provides a uniform decelerating field between itself, andthe photo conductive layer, so that the electron beam approachesthe layer with a low velocity to prevent any secondary emission.

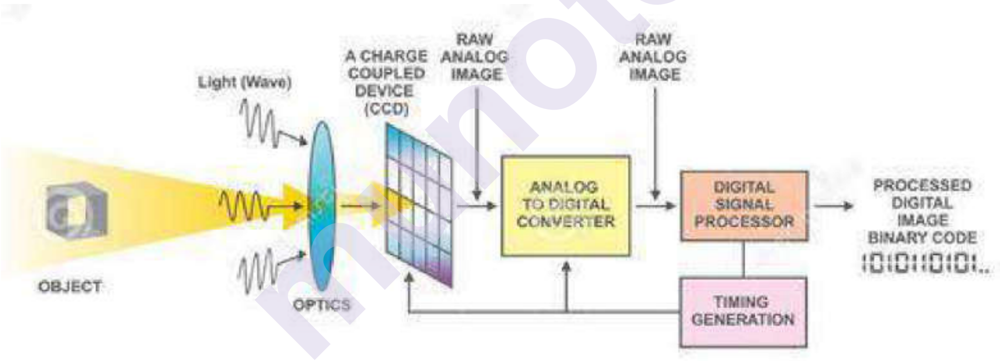
The fluctuating voltage coupled out to a video amplifier can be usedto reproduce the target.

***Digital camera***

A digital camera is a camera that captures images and turns them into digital form.

Digital camera shares an optical system which uses a lens with a variable diaphragm to focus light onto an imagepickup device.

The diaphragm and shutter admit the correct amount oflight to the imager.




Digital camera contains image sensors that captures theincoming light rays and turns them into electrical signals.

This image sensors can be of two types- i) charge-coupled device (CCD) or ii)CMOS image sensor.

Light from the object zooms into the camera lens.

This incoming light hit the image sensor, which breaks it upinto millions of pixels.

The sensor measures the color and brightness of each pixeland stores it as a number.

The output digital photograph is effectively a long string ofnumbers describing the exact details of each pixel itcontains.

# 1.9 COLOUR IMAGE FUNDAMENTALS

## 1.9.1 RGB

In the RGB model, an image consists of three independent image planes, one in each of the primary colors: red, green and blue. (The standard wavelengths for the three primaries are as shown in figure). Specifying a particular color is by specifying the amount of each of the primary components present. Figure 1.21 shows the geometry of the RGB color model for specifying colors using a Cartesian coordinate system. The grayscale spectrum, i.e. those colors made from equal amounts of each primary, lies on the line joining the black and white vertices.

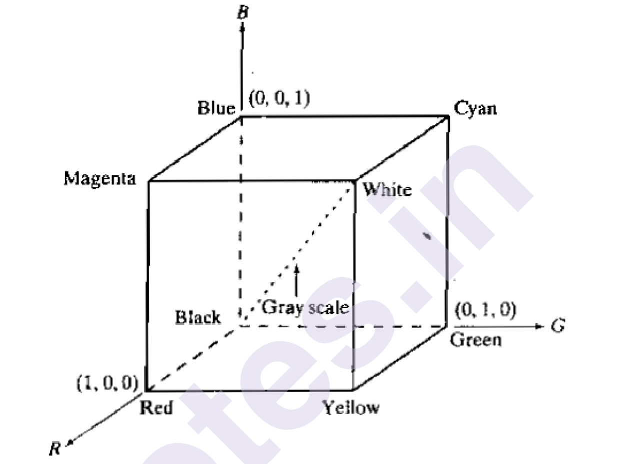



Fig 1.21 Schematic of the RGB color cube

Fig.1.21 The RGB color cube. The gray scale spectrum lies on the line joining the black and white vertices.

This is an additive model, i.e. the colors present in the light add to form new colors, and is appropriate for the mixing of colored light for example. The image on the left of figure 1.22 shows the additive mixing of red, green and blue primaries to form the three secondary colors yellow (red + green), cyan (blue + green) and magenta (red + blue), and white ((red + green + blue). The RGB model is used for color monitors and most video cameras.

Fig.6.2 RGB 24 bit color cube

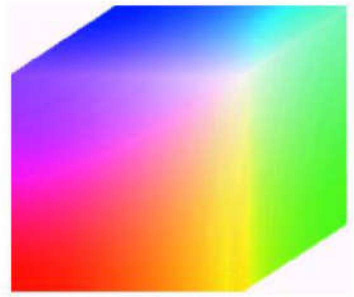
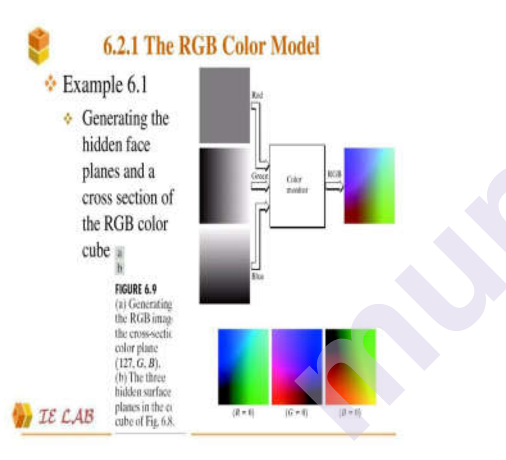
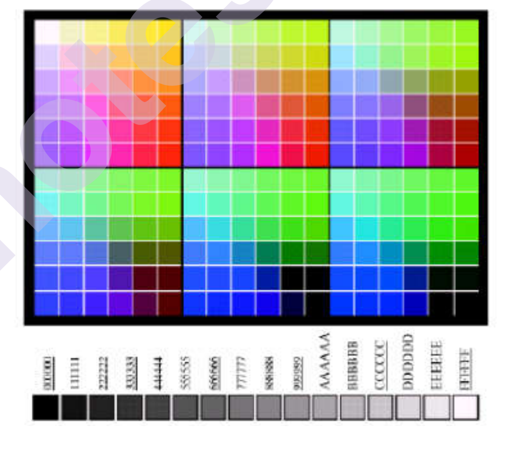


Fig. 1.22 24-bit color cube

Fig.1.23 The figure on the left shows the additive mixing of red, green and blue primaries to form the three secondary colors yellow (red + green), cyan (blue + green) and magenta (red + blue), and white (red + green + blue). The figure on the right shows the three subtractive primaries and their pairwise combinations to form red, green and blue, and finally black by subtracting all three primaries from white.

<table>
<tr><td>
</td><td>
</td></tr>
<tr><td>Fig 1.23 generating the RGB image of the cross sectional color plane</td><td>Fig. 1.24 safe 216 RGB colors and gray in 256-color RGB system</td></tr>
</table>



***Pixel Depth:***

The number of bits used to represent each pixel in the RGB space is called the pixel depth. If the image is represented by 8 bits then

the pixel depth of each RGB color pixel = 3*number of bits/plane=3*8=24

A full color image is a 24 bit RGB color image. Therefore total number of colors in a full color image = $(28)^3$ = 16,777,216

***Safe RGB colors:***

Most of the system use 256 colors. Withoutt depending on the hardware capabilities of the system the system reproduces subset of colors which is called the set of RGB colors or the set of all systems safe colors.

***Standard safe colors:***

It is assumed that a minimum number of 256 colors can be reproduced by any system. Among these, 40 colors are found to be processed differently by different operating system. The remaining 216 colors are called as standard safe colors.

***Component values of safe colors:***

Each of the 216 safe colors can be formed from three RGB component values. But each component value should be selected only from the set of values {0, 51,102,153,204,255}, in which the successive numbers are obtained by adding 51 and are divisible by 3 therefore total number of possible values= 6*6*6=216

**Hexadecimal representation**

The component values in RGB model should be represented using hexadecimal number system. The decimal numbers 1,2,….14,15 correspond to the hex numbers 0,1,2,….9,A,B,C,D,E,F. the equivalent representation of the component values is given in table:

| Number System | Color Equivalents | | | | | |
|---|---|---|---|---|---|---|
| Hex | 00 | 33 | 66 | 99 | CC | FF |
| Decimal | 0 | 51 | 102 | 153 | 204 | 255 |

**Applications:**

Color monitors, Color video cameras

Advantages:

- Image color generation
- Changing to other models such as CMY is straight forward
- It is suitable for hardware implementation
- It is based on the strong perception of human vision to red, green andblue primaries.

**Disadvantages:**

- It is not acceptable that a color image is formed by combining three primary colors.

- This model is not suitable for describing colors in a way which is practical for human interpretation.

### 1.9.2 CMY

The CMY (Cyan Mmagenta Yellow) model is a *subtractive* model appropriate to absorption of colors. The CMY model asks what is subtracted from white. The primary colors are cyan, magenta and yellow, and secondary colors are with red, green and blue

The surface coated with cyan pigment is illuminated by white light, no red light is reflected, and similarly for magenta and green, and yellow and blue. The relationship between the RGB and CMY models is given by:

$$\begin{matrix} C \\ M \\ Y \end{matrix} = \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} - \begin{matrix} R \\ G \\ B \end{matrix}$$

The CMY model is used by printing devices and filters.

### 1.9.3 HIS MODELS

Colors are specified by the three quantities hue, saturation and intensity which is similar to the way of human interpretation.

Hue: It is a color attribute that describes a pure color.

Saturation: It is a measure of the degree to which a pure color is diluted by white light.

Intensity: It is a measureable and interpretable descriptor of monochromatic images, which is also called the gray level.

*i) Hue:*

The hue of a color can be determined from the RGB color cube. if the three points black, white and any one color are combined, a triangle is formed. All the points inside the triangle will have the same hue. This is due to the fact that black and white components cannot change the hue.

*HSI color space*

The HIS color space is represented by vertical intensity axis and locus of color points that lie on planes perpendicular to the axis. The shape of the cube is defined by the intersecting points of these planes with the faces of cube. As the planes move up and down along the intensity axis, the shape can either be a triangle or a hexagon. In HSI space, primary colors are separated by 120°. Secondary colors are also separated by 120° and the angle between the secondary's and primaries' is 60°.

***Representation of Hue:***

The hue of a color point is determined by an angle from some reference point.

The angle between the point and the red axis is 0° is zero hue.

If the angle from red axis increases in the counter clock wise direction then hue increases.

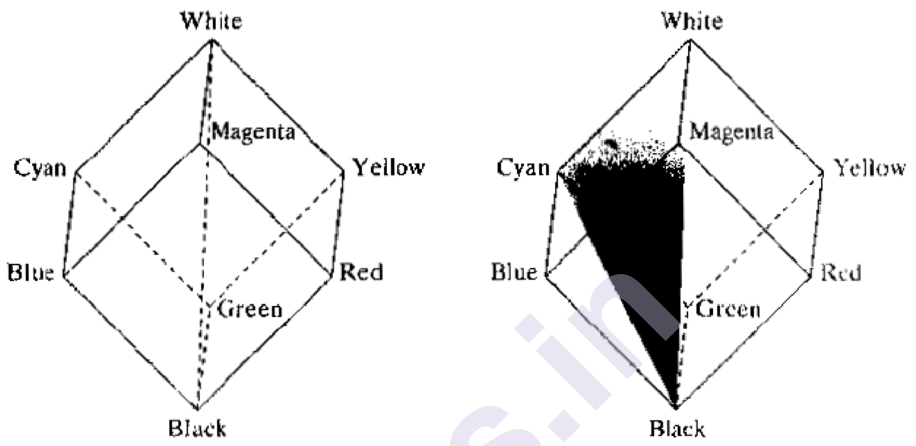



Fig. 1.25 Conceptual relationship between the RGB and HSI color models

***ii) Intensity:***

The intensity can be extracted from an RGB image because an RGB color image is viewed as three monochrome intensity images.

***Intensity Axis:***

A vertical line joining the black vertex (0, 0, 0) and white vertex(1,1, 1) is called intensity axis. The intensity axis represents the gray scale.

***iii)saturation:***

All points on the intensity axis are gray which means that the saturation i.e., purity of points on the axis is zero.

When the distance of a color from the intensity axis increases, the saturation of that color also increases.

*Representation of saturation*

The saturation is described as the length from the vertical axis.

In the HSI space, it is represented by the length of the vector from the origin to the color point.

If the length is more the saturation is high and vice versa.

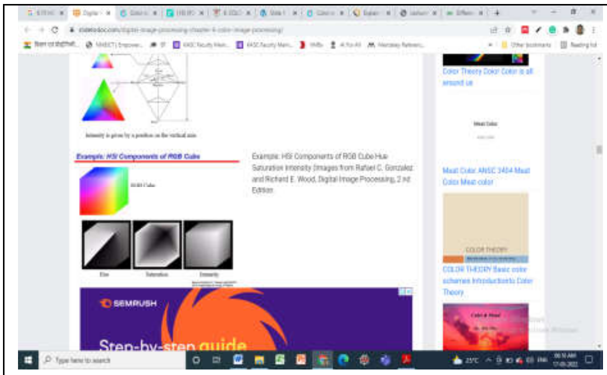



Fig.1.26 HIS Components of the images

## Converting colors from RGB to HSI

Given an image in RGB color format the H component of each RGB pixel is obtained usin the equation

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{cases} \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{cases}$$

Saturation component is

S = 1- QUOTE [min(R, G, B)]

Intensity component is

I = QUOTE (R + G + B)



Converting colors from HSI to RGB

Converting equations depend on the value of H (H – Hue)> for three sectors the equation for conversion is given below:

RG (Red, Green) Sector ($0^{\circ} \leq H < 120^{\circ\circ} \leq H < 120^{\circ}$) : When H is in this sector the RGB components are given by the equations

B = I(1-S)

G = 3I - (R + B)

GB (Green, Blue) Sector (120$^{\circ} \leq H < 240^{\circ\circ} \leq H < 240^{\circ}$) : When H (H=H-120$^{\circ}$) $^{\circ}$) is in this sector the RGB components are given by the equations:

$$R = I(1-S)$$

$$B=3I - (R + G)$$

BR (Blue, Red) Sector (240$^{\circ} \leq H < 360^{\circ\circ} \leq H < 360^{\circ}$) : When H (H=H-240$^{\circ}$) $^{\circ}$) is in this sector the RGB components are given by the equations:

$$G = I(1-S)$$

$$R =3I - (G + B)$$

**Advantages of HSI model:**

- It describes colors in terms that are suitable for human interpretation.
- The model allows independent control over the color describing quantities namely hue, saturation and intensity.
- It can be used as an ideal tool for developing image processing algorithms based on color descriptions.

### 1.9.4 2D SAMPLING

To create a digital image, convert the continuous sensed data into digital form. This involves two processes.

i) Sampling

ii) Quantization

An image, f(x, y), may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, sample the function in both coordinates and in amplitude.

Digitizing the coordinate values is called Sampling.

The one-dimensional function in Fig. 1.27(b) is a plot of amplitude (intensity level) values of the continuous image along the line segment AB in Fig. 1.27 (a).

To sample this function, equally spaced samples along line AB, are depicted in Fig. 1.27 (c). The spatial location of each sample is indicated by a vertical tick mark.

The samples are shown as small white squares super imposed on the function. The set of these discrete locations gives the sampled function.

However, the values of the samples still span (vertically) a continuous range of intensity values.

The intensity values must be (quantized) to form a digital function

The right side of Fig. 1.27 (c) shows the intensity scale divided into eight discrete intervals, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight intensity intervals. The continuous intensity levels are quantized by assigning one of the eight values to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig1.27 (d). Starting at the top of the image and carrying out this procedure line by line produces a two-dimensional digital image.

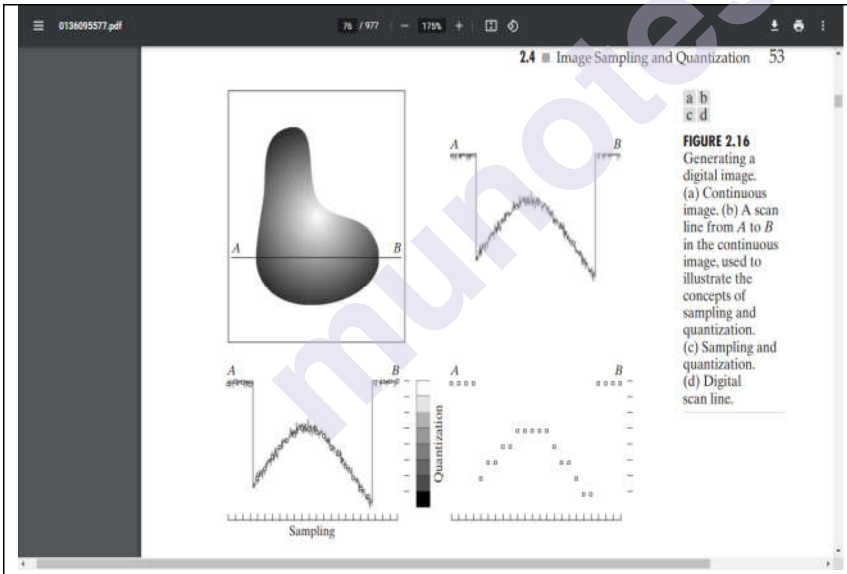



Fig. 1.27 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



### 1.9.5 QUANTIZATION

Digitizing the amplitude values is called Quantization.

Quantisation involves representing the sampled data by a finite number of levels based on some criteria such as minimisation of quantiser distortion.

Quantisers can be classified into two types, namely, i) scalar quantisers and ii) vector quantisers. The classification of quantisers is shown in Fig. 1.29.

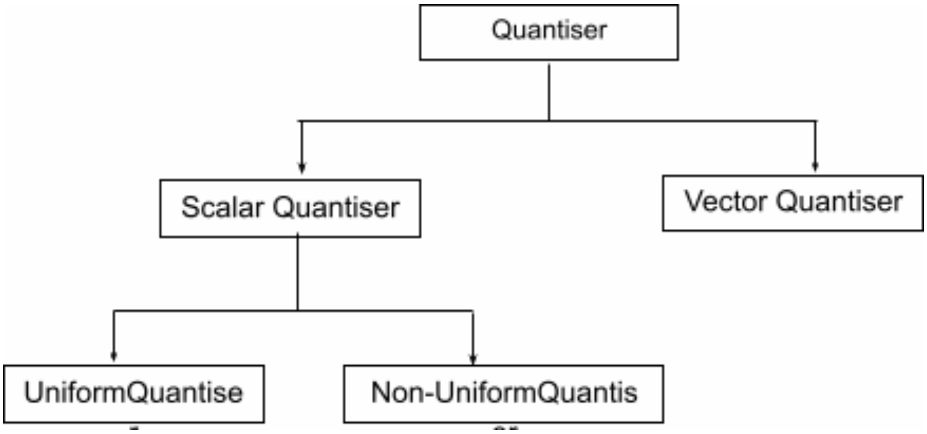



Fig. 1.29 Classification of Quantiser

Selecting the number of individual mechanical increments for spatial sampling at which the sensor to collect data for activation. Limits on sampling accuracy are determined by the factors, such as the quality of the optical components of the system.

Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other.

The accuracy achieved in quantization is highly dependent on the noise content of the sampled signal. The method of sampling is determined by the sensor arrangement used to generate the image.

When an image is generated by a single sensing element combined with mechanical motion, and then the output of the sensor is quantized as given in Fig. 2.18.

The image after sampling and quantization is shown in fig 2.18 (b). The quality of a digital image is determined to a large degree by the number of samples and discrete intensity levels used in sampling and quantization.

## 1.10 SUMMARY

Since1921 when the Bartlane cable picture transmission system was introduced the Digital images started its evolution. In 1964 the computers are used to process digital images and the actual digital image processing started working.

Digital image composed of elements called pixels. For the immediate output display, fast processing and huge storage the digital images are used.

The position of the pixels to represent the digital images are identified through neibors of the pixels, adjacency, boundaries and connectivities of the pixels.

Image Acquisition devices, Image storage devices, Image processing elements and Image display devices are the basic elements of the digital image processing sytem which are used to process the digital images. The structure of human eye helps the human to understand and sense the colors and structure of the images.

RGB , CMY are useful in representing the images with different colors, brightness and contrasts.

## 1.11 REFERENCES


1. R.C.Gonzalez&R.E.Woods, Digital Image Processing, Pearson Education, 3rd edition, ISBN. 13:978-0131687288
2. S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN- 13:978-0-07- 0144798
3. William K. Pratt, "Digital Image Processing", John Wiley, NJ, 4th Edition,2007
4. The Origins of Digital Image Processing & Application areas in Digital Image Processing Medical Images,Mukul, Sajjansingh, and Nishi


## 1.12 UNIT END EXERCISES

1. Define Image and Digital Image
2. Classify the images.
3. Write the advantages and disadvantages of digital images.
4. What is digital image processing?
5. How do you represent the digital Images? Explain
6. Describe the relationship between pixels
7. How do measure the distance between pixels?
8. Explain the elements of digital image processing system.
9. Explain the structure of human eye
10. Write short note on i) Hue ii)Mach band effect
11. Elucidate the working principles of digital camera with neat diagram
12. Write short note on i) RGB ii)CMY

❖❖❖❖

# Module II

# Image Enhancement in the spatial domain

# 2

# SPATIAL DOMAIN METHODS

**Unit Structure**



## 2.0 OBJECTIVES

Enhancement's main goal is to improve the quality of an image so that it may be used in a certain process.

- Enhancement of images Enhancement in the spatial domain and Frequency domain fall into two categories.
- The word spatial domain refers to the Image Plane itself, which is DIRECT pixel manipulation.
- Frequency domain processing approaches work by altering an image's Fourier transform.

## 2.1 INTRODUCTION

The aggregate of pixels that make up an image is known as the spatial domain.

Spatial Domain Methods are procedures that work on these pixels directly.

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

F(x,y): Input Image, T: Image Operator g(x,y): Image that has been processed.

T can also work with a group of images.

The neighborhood is defined as:

Input for Process: A one-pixel neighborhood around a point (x,y) The most basic kind of input is a one-pixel neighborhood. s=T(r) T:Transformation Function s,r: f(x,y) and g(x,y) grey levels, respectively. The most basic technique is a rectangular sub-picture region centred at (x,y).

- **SPATIAL DOMAIN METHODS**

*The value of a pixel in the enhanced picture with coordinates (x,y) is the outcome of executing some operation on pixels in the vicinity of (x,y) in the input image, F.*

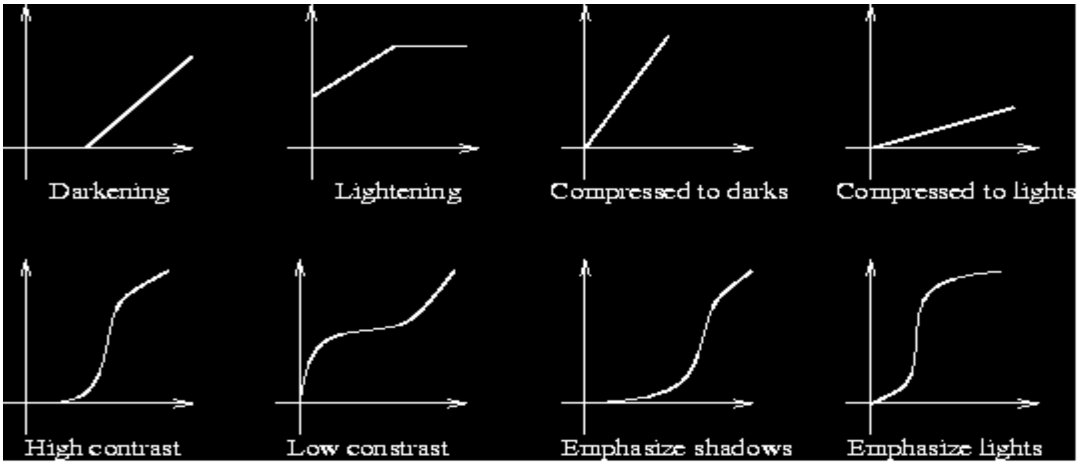
*Neighbourhoods can be any shape, however they are most commonly rectangular.*

- ***GREY SCALE MANIPULATION***

When the operator T just acts on a pixel neighborhood in the input image, it is the simplest kind of an operation because it only depends on the value of F at that point (x,y). This is a greyscale mapping or transformation.

Thresholding is the simplest case, in which the intensity profile is replaced with a step function that is active at a set threshold value. In this scenario, any pixel in the input image with a grey level below the threshold is mapped to 0 in the output image. The rest of the pixels are set to 255.

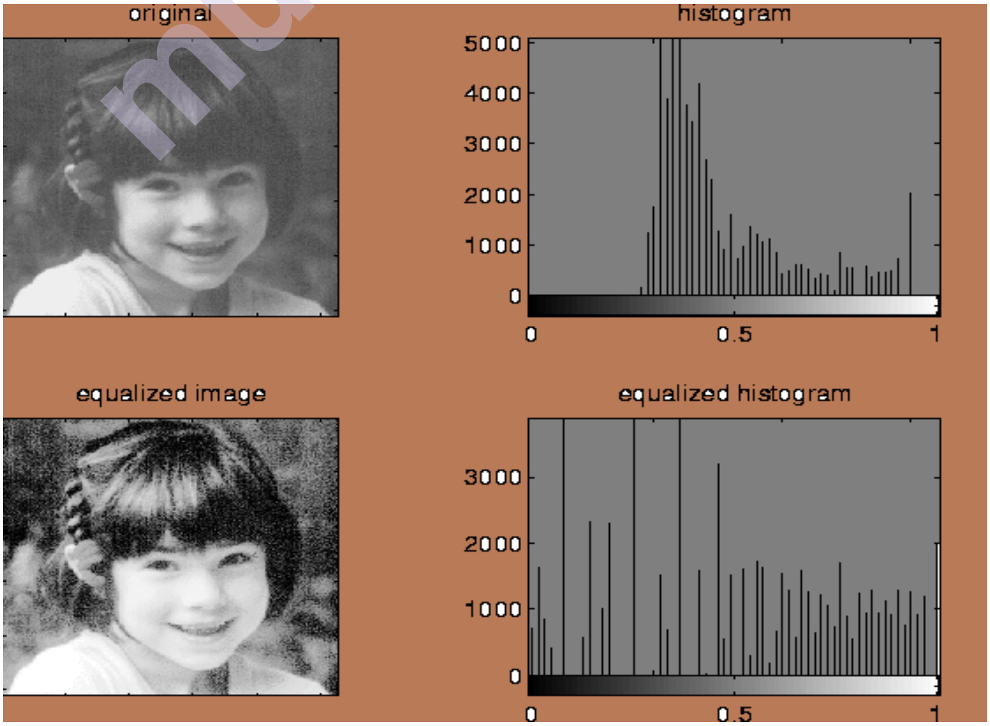
Figure 1 depicts further greyscale adjustments.




- **EQUALIZATION OF HISTOGRAMS**

Equalization of histograms is a typical approach for improving the appearance of photographs. Assume that we have a largely dark image. The visual detail is compressed towards the dark end of the histogram, and the histogram is skewed towards the lower end of the greyscale. The image would be much clearer if we could stretch out the grey levels at the dark end to obtain a more consistently distributed histogram.

Figure 2 shows the original image, histogram, and equalised versions. Both photos have been quantized to a total of 64 grey levels.



Finding a grey scale translation function that produces an output image with a uniform histogram is the goal of histogram equalisation (or nearly so).

What is the procedure for determining the grey scale transformation function? Assume that our grey levels are continuous and that they have been normalised to a range of 0 to 1.

We need to identify a transformation T that converts the grey values r in the input image F to grey values s = T(r) in the converted image.

The assumption is that

- T is single valued and monotonically increasing, and
- $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.

The inverse transformation from $s$ to $r$ is given by

$r = T^{1}(s)$.

We have a probability distribution for grey levels in the input image Pr if we take the histogram for the input image and normalise it so that the area under the histogram is Pr(r).

What is the probability distribution Ps(s) if we transform the input image to s = T(r)?

It turns out that, according to probability theory,

$$P_s(s) = P_r(r)\frac{dr}{ds},$$

where $r = T^{1}(s)$.

Consider the transformation

$$s = T(r) = \int_0^r P_r(w)dw.$$

The cumulative distribution function of r is represented by this. The derivative of s with respect to r is calculated using this definition of T.

$$\frac{ds}{dr} = P_r(r).$$

Substituting this back into the expression for $P_s$, we get

$$P_s(s) = P_r(r)\frac{1}{P_r(r)} = 1$$

for all $s, where 0 \leq s \leq 1$ .Thus, $P_s(s)$ is now a uniform distribution function, which is what we want.

- **DISCRETE FORMULATION**

The probability distribution of grey levels in the input image must first be determined. Now

$$P_r(r) = \frac{n_k}{N}$$

where $n_k$ is the number of pixels having grey level k, and $N$ is the total number of pixels in the image.

The transformation now becomes

$$s_k = T(r_k) = \sum_{i=0}^{k} \frac{n_i}{N}$$
$$= \sum_{i=0}^{k} P_r(r_i).$$

Note that $0 \leq r_k \leq 1$,

the index $k = 0, 1, 2, \ldots, 255$ ,and $0 \leq s_k \leq 1$.

*So that the output values of this transformation span from 0 to 255, the values of* $S_k$ *must be scaled up by 255 and rounded to the nearest integer. As a result of the discretization and rounding of* $S_k$ *to the nearest integer, the modified image's histogram will not be exactly uniform.*

- **SMOOTHING AN IMAGE**

Image smoothing is used to reduce the impact of camera noise, erroneous pixel values, missing pixel values, and other factors. Image smoothing can be done in a variety of ways; we'll look at neighborhood averaging and edge-preserving smoothing.

- **NEIGHBOURHOOD AVERAGING**

The average pixel value in a neighbourhood of $\hat{F}(x, y)$ is obtained from the average pixel value in a neighbourhood of $(x,y)$ in the input image.

For example, if we use a $3 \times 3$ neighbourhood around each pixel we would use the mask

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

Each pixel value is multiplied by 1/9and then totalled before being placed in the resulting image. This mask is moved across the image in steps until every pixel is covered. This soothing mask is used to convolve the image (also known as a spatial filter or kernel).

The value of a pixel, on the other hand, is normally expected to be more strongly related to the values of pixels nearby than to those further away. This is because most points in a picture are spatially coherent with their neighbours; in fact, this hypothesis is only false at edge or feature points. As a result, the pixels towards the mask's center are usually given a higher weight than those on the edges.

The rectangular weighting function (which just takes the average over the window), a triangular weighting function, and a Gaussian are all typical weighting functions.

Although Gaussian smoothing is the most widely utilized, there isn't much of a difference between alternative weighting functions in practice. Gaussian smoothing is characterized by the smooth modification of the image's frequency components.

Smoothing decreases or attenuates the image's higher frequencies. Other mask shapes can cause strange things to happen to the frequency spectrum, but we normally don't notice much in terms of image appearance.

Smoothing that preserves the edge

Because the image's high frequencies are suppressed, neighborhood averaging or Gaussian smoothing will tend to blur edges. Using median filtering as an alternative is a viable option. The grey level is set to the median of the pixel values in the pixel's immediate vicinity.

The median m of a set of values is the value at which half of the values are less than m and the other half are greater. Assume that the pixel values 3 x 3 in a given neighborhood are (10, 20, 20, 15, 20, 20, 20, 25, 100). We obtain (10, 15, 20, 20, |20|, 20, 20, 25, 100) if we order the values, and the median is 20.

The result of median filtering is that pixels with outlying values are forced to become more like their neighbors while maintaining edges. Median filters, by definition, are non-linear.

Median filtering is a morphological operation. Pixel values are replaced with the smallest value in the neighborhood when we erode an image.

When distorting an image, the greatest value in the neighborhood is used to replace pixel values. Median filtering replaces pixels with the neighborhood's median value. The type of morphological operation is determined by the rank of the value of the pixel used in the neighborhood.

Figure 3: Image of Genevieve with salt and pepper noise, averaging result, and median filtering result.




## 2.2 AN OVERVIEW

The spatial domain technique is a well-known denoising technique. It's a noise-reduction approach that uses spatial filters to apply directly to digital photos. Linear and nonlinear spatial filters are the two types of spatial filtering algorithms (Sanches et al., 2008). Filtering is a method used in image processing to do several preprocessing and other tasks such as interpolation, resampling, denoising, and so on. The type of task performed by the filter method and the type of digital image determine the filter method to be used. Filter methods are used in digital image processing to remove undesirable noise from digital photographs while maintaining the original image (Priya et al., 2018; Agostinelli et al., 2013).

Nonlinear filters are used in a variety of ways, the most common of which is to remove a certain sort of unwanted noise from digital photographs. There is no built-in way for detecting noise in the digital image with this method. Nonlinear filters often eliminate noise to a certain point while blurring images and hiding edges. Several academics have created various sorts of median (nonlinear) filters to solve this challenge throughout the previous decade. The median filter, partial differential equations, nonlocal mean, and total variation are the most used nonlinear filters. A linear filter is a denoising technique in which the image's output results vary in a linear fashion. Denoising outcomes are influenced by the image's input. As the image's input changes, the image's output changes linearly. The processing time of linear filters for picture denoising is determined by the input signals and the output signals. The mean linear filter is the most effective filter for removing Gaussian noise from digital medical pictures. This approach is a simple way to denoise digital photos (Wieclawek and Pietka, 2019). The average or mean pixels values of the neighbour pixels are calculated first, and then replaced with every pixel of the digital image in the mean filter. To reduce noise from a digital image, it's a very useful linear filtering approach. Wiener filtering is another linear filtering technique. This technique requires all additive noise, noise spectra, and digital picture inputs, and it works best if all of the input signals are in good working order. This strategy reduces the mean square error of the intended and estimated random processes by removing noise.

## 2.3 SPATIAL DOMAIN METHODS

For image enhancement, there are primarily two methods: one for images in the spatial domain and the other for images in the frequency domain. The first method is based on editing individual pixels in an image, whereas the second way is based on altering an image's Fourier transform.

**Spatial domain methods**

Here, image processing functions can be expressed as :

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

f(x,y) is the input picture, g(x,y) is the processed image (i.e. the result or output image), and an operator on f is defined over some neighbourhood N of (x,y). We usually employ a rectangle subimage centred at N for (x,y).

**a) N is a 1×1 neighbourhood (point-processing)**

N encompasses exactly one pixel in this case. The operator T is then transformed into a gray-level transformation function, which is written as:

$$s = T(r),$$

The gray levels of f(x,y) and g are represented by r,s (x,y). We can produce some intriguing effects with this technique, such as contrast stretching and bi-level mapping (here an image is converted so that it only contains black and one color white). The challenge is to define T in such a way that it darkens grey levels below a particular threshold k and brightens grey levels above it. A black-and-white image is created when the darkening and brightening are both consistent (black and white). This technique is known as 'point-processing' since s is only dependent on the value (i.e. the gray-level) of T in a single pixel.

**b) N is a m×m neighbourhood (spatial filtering)**

In this situation, N refers to a small area. It's worth noting that this technology isn't limited to image enhancement; it can also be used to smoothen photos, among other things. The values in a predefined neighborhood (i.e. the mask/filter) of g(x,y) are used to determine the value of g(x,y) (x,y). The value of m can range from 3 to 10 in most cases. These procedures are known as mask processing' or 'filtering.'

**METHODS IN THE FREQUENCY DOMAIN**

The convolution theorem is at the heart of these techniques. The following is an example of what it means:

Assume that g(x,y) is a convolution of an image f(x,y) and a linear, position invariant operator h(x,y):

$$g(x,y) = h(x,y) * f(x,y).$$

Applying the convolution theorem yields :

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

The Fourier transforms of f, g, and h are F, G, and H, respectively. The following is the result of applying the inverse Fourier transform to G(u,v):

$$g(x,y) = \mathcal{F}^{-1}(H(u,v) \cdot F(u,v))$$

H(u,v), for example, enhances the high-frequency components of f(u,v), resulting in a g(x,y) picture with exaggerated edges.

Some intriguing features can be noticed when looking at the theory of linear systems (see figure 1): A system with the function of producing an out-put image g(x,y) from an input image f (x,y) is referred to as h(x,y). The Fourier notation for this operation is equivalent to this.

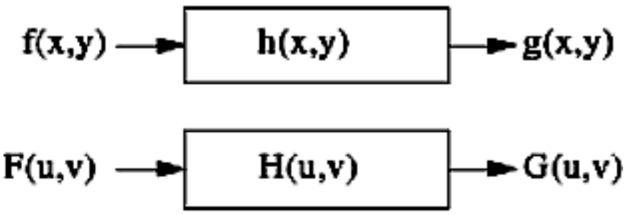



**Figure 1**: Linear systems.

## 2.3.1 POINT PROCESSING

When making a film, it's common to lessen the overall intensity to create a unique atmosphere. Some people go overboard, and the effect is that the observer can only see blackness. So, what exactly do you do? You take out your remote and press the brightness button to alter the light intensity. When you do this, you're performing a type of image processing called point processing.

Let's say we have an input image f(x, y) that we want to alter to get a different image, which we'll call the output image g. (x,y). When altering the brightness of a movie, the input picture is the one saved on the DVD you're watching, and the output image is the one that appears on the television screen. Point processing is now described as an operation that calculates the new value of a pixel in g(x, y) based on the value of the same pixel in f(x, y) and some action. That is, in f(x, y), the values of a pixel's neighbours have no influence, hence the name point processing. The adjacent pixels will play a significant role in the upcoming subjects. Figure 4.1 depicts the principle of point processing. Some of the most fundamental point processing operations are explained in this topic.

When you use your remote to adjust the brightness, you're actually changing the value of b in the following equation:

$$g(x, y) = f(x, y) + b \tag{4.1}$$

The value of b is increased every time you press the '+' brightness button, and vice versa. As b is increased, a higher and higher value is added to each pixel in the input image, making the image brighter. The image becomes brighter if b > 0, and darker if b 0. Figure 2.2 depicts the effect of altering the brightness.

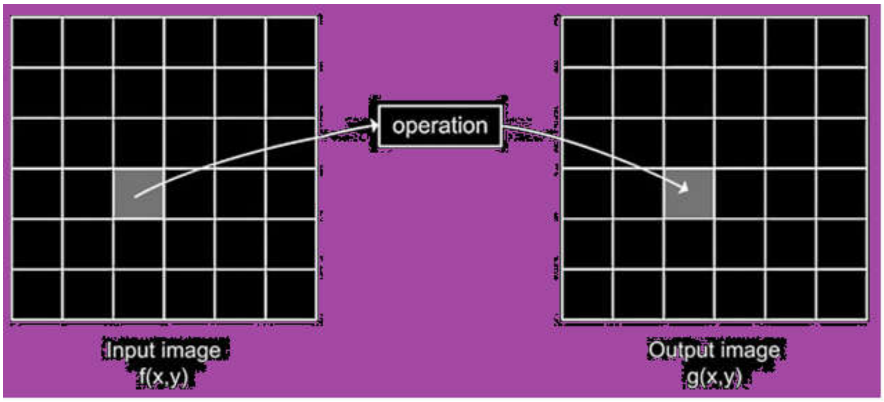


Figure 2.1: The point-processing principle. A pixel in the input image is processed, and the result is saved in the output image at the same location.

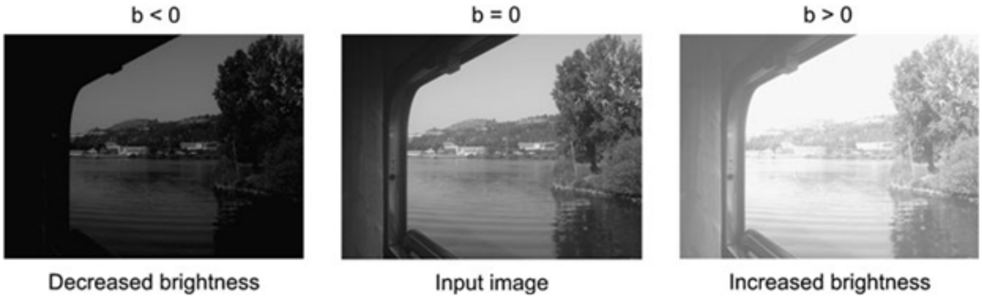


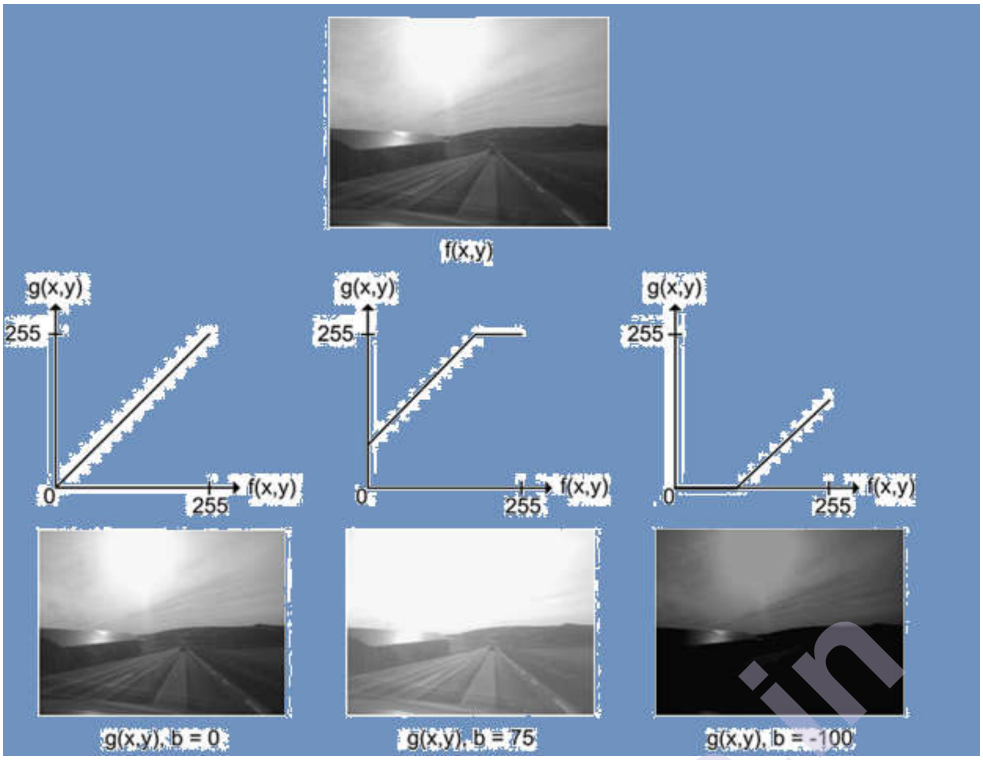

Figure 2.2: The resultant image will be equivalent to the input image if b in Eq. 2.1 is zero. If b is a negative quantity, the image produced will be smaller.

If b is a positive number, the brightness of the resulting image will be increased.

The use of a graph, as shown in Fig. 2.3, is often a more convenient manner of illustrating the brightness action. The graph depicts the mapping of pixel values in the input image (horizontal axis) to pixel values in the output picture (vertical axis) (vertical axis). Gray-level mapping is the name given to such a graph. The mapping does nothing in the first graph, i.e., g(142,42) = /. (142,42).

In the following graph, all pixel values are increased (b > 0), resulting in a brighter image. This has two effects: I no pixel in the output image will be fully dark, and ii) some pixels in the output image will have a value greater than 255. The latter is undesirable due to an 8-bit image's upper limit, hence all pixels above 255 are set to 255, as shown in the graph's horizontal section. When b 0 is set to zero, some pixels will have negative values and will be set to zero in the output, as shown in the previous graph.

You can adjust the contrast in the same way that you can adjust the brightness on your TV. The gray-level values that make up an image's contrast are how distinct they are. When we look at two pixels with values 112 and 114 adjacent to each other, the human eye has trouble distinguishing them, and we remark there is a low contrast. If the pixels are 112 and 212, on the other hand, we can readily differentiate them and claim the contrast is great.




Three instances of gray-level mapping are shown in Figure 2.3. The input is shown at the top of the page. The three additional images are the result of the three gray-level mappings being applied to the input. Eq. 4.1 is used in all three gray-level mappings.


Figure 2.4: If the value of an in Eq. 2.2 is one, the output image will be the same as the input image. If an is less than one, the resulting image will be less contrasted; if an is greater than one, the resulting image will be more contrasted.

Changing the slope of the graph1 changes the contrast of an image:

$$g(x, y) = a \cdot f(x, y)$$

If an is more than one, the contrast is raised; if it is less than one, the contrast is diminished. When a = 2, the pixels 112 and 114, for example, will have the values 224 and 228, respectively. The contrast is raised by a factor of two because the difference between them is increased by a factor of two. The effect of adjusting the contrast may be observed in Fig. 4.4.

When the equations for brightness (Eq. 2.1) and contrast (Eq. 2.2) are combined, we get

$$g(x, y) = a \cdot f(x, y) + b$$

Which is a straight line's equation. Consider an example of how to use this equation. Let's say we're interested in a section of the input image where the contrast isn't quite right. As a result, we determine the range of pixels in this region of the image and map them to the complete [0, 255] range in the output image. Assume that the input image's minimum and maximum pixel values are 100 and 150, respectively.

Changing the contrast implies that in the output image, all pixel values below 100 are changed to zero, and all pixel values above 150 are set to 255. Eq. 2.3 is used to map the pixels in the range [100, 150] to [0, 255], where a and b are defined as follows:

$$a = \frac{255}{f_2 - f_1}, \qquad b = -a \cdot f_1$$

Non-linear Gray-Level Mapping

Gray-level mapping isn't confined to Eq. 2.3-defined linear mappings. In fact, the designer is free to specify the gray-level mapping as she wants as long as each input value has just one output value. Rather than creating a new equation/graph, the designer will frequently use one that is already defined. The following are three of the most frequent non-linear mapping functions.

Gamma Mapping

It is the process of converting one colour into another.

Because humans have a non-linear sense of contrast, it is useful to be able to adjust the contrast in the dark grey levels and the light grey levels separately in various cameras and display devices (for example, flat panel televisions). Gamma mapping is a typical non-linear mapping that is defined for positive as

$$g(x, y) = f(x, y)^{\gamma}$$

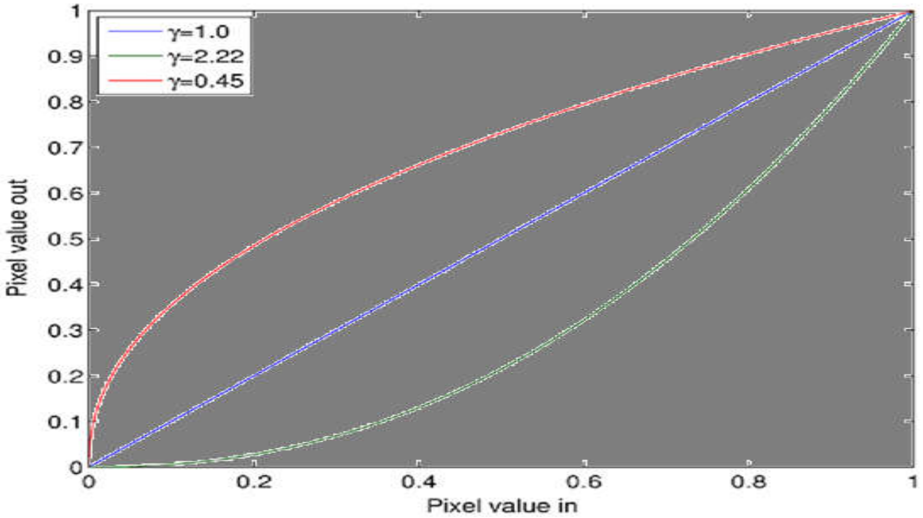



Fig. 4.5 Curves of gamma-mapping for various gammas

Figure 2.5 depicts a few gamma-mapping curves. We get the identity mapping if = 1. We boost the mid-levels for 0 1 to increase the dynamics in the dark sections. We decrease the mid-levels to increase the dynamics in the bright areas for > 1. The gamma mapping is set up so that both the input and output pixel values are between 0 and 1. Before applying the gamma transformation, the input pixel values must first be transformed by dividing each pixel value by 255. After the gamma transformation, the output values should be scaled from [0, 1] to [0, 255].

There is a specific case presented. A pixel with the value vin = 120 in a gray-scale picture is gamma mapped with = 2.22. Initially, the pixel value is divided by 255 to convert it to the interval [0,1], v = 120/255 = 0.4706. Second, v2 = 0.47062.22 = 0.1876 is used to do gamma mapping. Finally, the result is vout = 0.1876 • 255 = 47, which is transferred back to the interval [0,255]. Figure 4.6 depicts some examples.


Figure 2.6: With a value of 0.45, gamma mapping to the left is 0.45, while with a value of 2.22, gamma mapping to the right is 2.22. The original image is in the middle.

**Mapping on a Logarithmic Scale**

The logarithm operator is used in an alternate non-linear mapping. The logarithm of the pixel value is used to replace each pixel. Low-intensity pixel values are amplified as a result of this. It's commonly employed when an image's dynamic range is too high to display or when there are a few bright spots on a dark background. Because there is no logarithm for zero, the mapping is defined as

$$g(x, y) = c \cdot \log\left(1 + f(x, y)\right) \tag{4.6}$$

Where c is a scaling constant that guarantees a maximum output value of 255 It is calculated as follows:

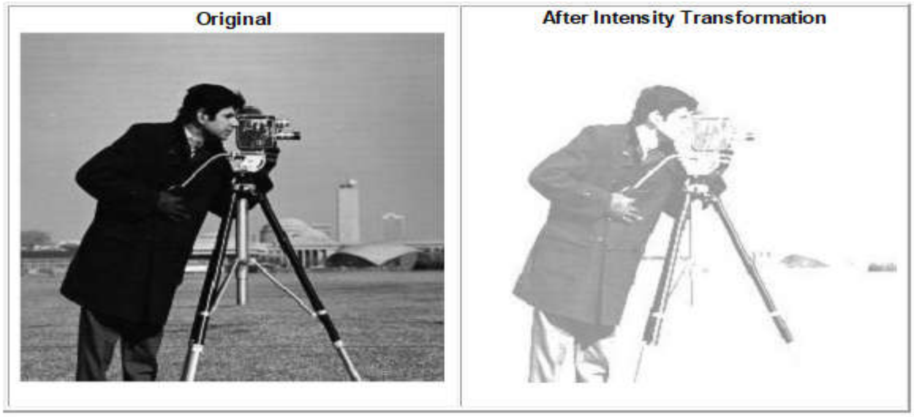
$$c = \frac{255}{\log(1 + v_{\max})} \tag{4.7}$$

Where umax is the input image's maximum pixel value. Changing the pixel values of the input image using a linear mapping before the logarithmic mapping can alter the behavior of the logarithmic mapping. Figure 4.7 shows the logarithmic mapping from [0,255] to [0,255]. This mapping will stretch low-intensity pixels while suppressing high-intensity pixels' contrast. Figure 4.7 shows one example.

## 2.3.2 INTENSITY TRANSFORMATIONS

When working with grayscale images, it's common to wish to change the intensity levels. For example, you might wish to flip the black and white intensities or make the darks darker and the lights lighter. Intensity modifications can be used to improve the contrast between various intensity values so that details in an image can be seen. The next two photos, for example, illustrate an image before and after an intensity modification.

The cameraman's jacket was originally black, but an intensity transformation enhanced the contrast between the black intensity values, which were previously too near, allowing the buttons and pockets to be seen. (This example is taken from the Image Processing Toolbox, User's Guide, Version 5 (MATLAB documentation)—found in the help menu or online at:

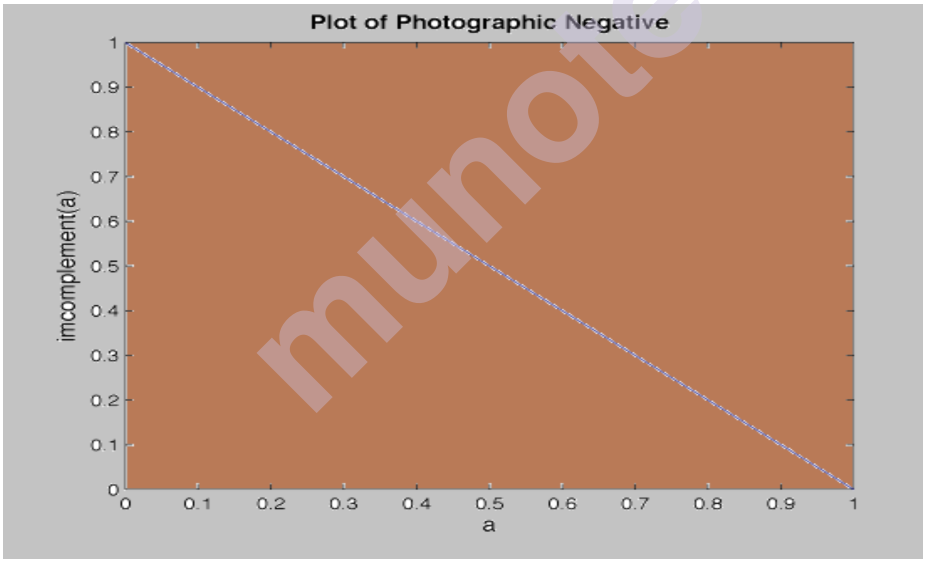


In general, Intensity Transformation Functions are used to adjust the intensity. The four main intensity transformation functions are discussed in the following sections:

1. photographic negative (using imcomplement)
2. gamma transformation (using imadjust)
3. logarithmic transformations (using c*log(1+f))
4. contrast-stretching transformations
(using 1./(1+(m./(double(f)+eps)).^E)

- **PHOTOGRAPHIC NEGATIVE**

The Photographic Negative is the most straightforward of the intensity conversions. Assume we're dealing with grayscale double arrays with black equal to 0 and white equal to 1. The notion is that 0s become 1s, and 1s become 0s, with any gradients in between reversed as well. This means that genuine black becomes true white and vice versa in terms of intensity. Incomplement provides a function in MATLAB that allows you to produce photographic negatives (f). The graph below displays the mapping between the original values (x-axis) and the incomplement function, with a=0:.01:1.




An example of a photography negative is shown below. Take note of how much easier it is to read the text in the middle of the tyre now than it was before:

| Original | Photographic Negative |
|---|---|
|  |  |

The MATLAB code that created these two images is:

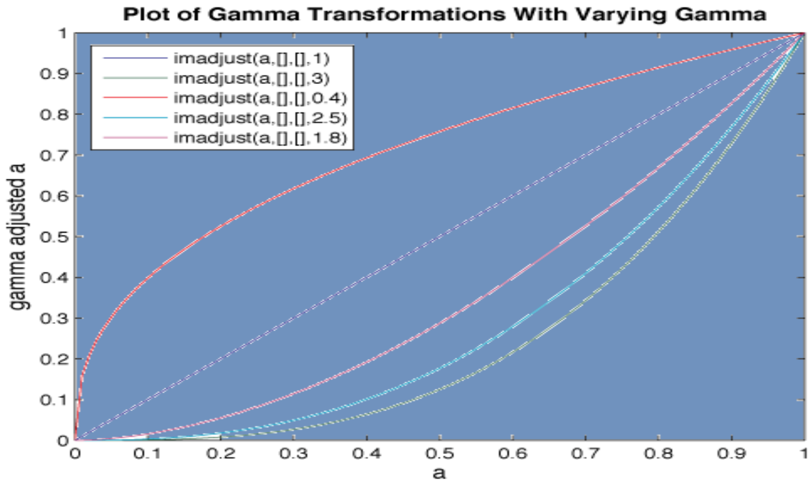
```
I=imread('tire.tif');

imshow(I)

J=imcomplement(I);

figure, imshow(J)
```

- **GAMMA TRANSFORMATIONS**

Gamma Transformations allow you to curve the grayscale components to brighten or darken the intensity (when gamma is less than one) (when gamma is greater than one). These gamma conversions are created using the MATLAB function:

imadjust(f, [low in high in], [low out high out], gamma) The input image is f, the curve is gamma, and the clipping is [low in high in] and [low out high out].

Values below low in and above high in are clipped to low out and high out, respectively. Both [low in high in] and [low out high out] are used in this lab with []. This indicates that the input's full range is mapped to the output's full range. The plots below show the effect of varying gamma with a=0:.01:1. Notice that the red line has gamma=0.4, which creates an upward curve and will brighten the image.




The outcomes of three of the gamma transformations indicated in the plot above are shown below. Notice how numbers greater than one result in a darker image, whilst values between 0 and one result in a brighter image with more contrast in dark places, allowing you to appreciate the tire's intricacies.

| **Original (and gamma=1)** | **gamma=3** | **gamma=0.4** |
|---|---|---|
|  |  |  |



**The MATLAB code that created these three images is:**

```
I=imread('tire.tif');

J=imadjust(I,[],[],1);

J2=imadjust(I,[],[],3);

J3=imadjust(I,[],[],0.4);

imshow(J);

figure,imshow(J2);

figure,imshow(J3);
```

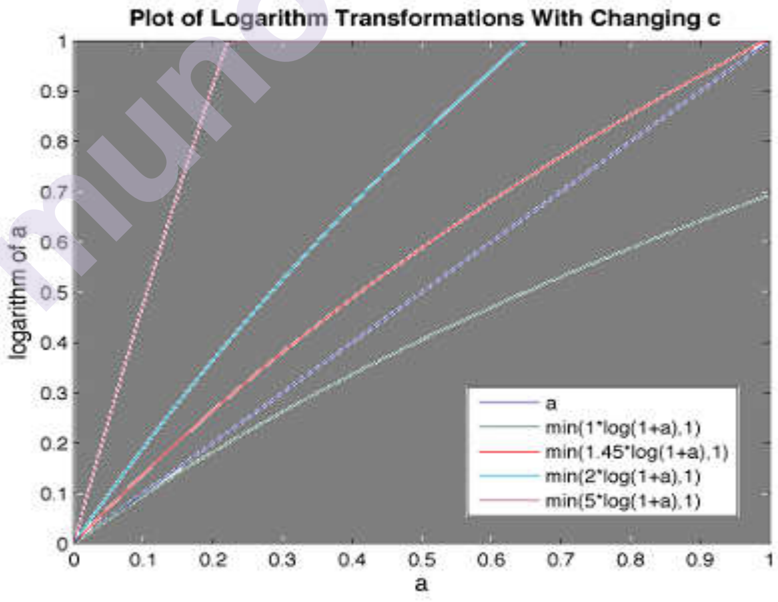
The gamma transformation is a crucial step in the image display process. You should find out more information about them. Charles Poynton, a digital video systems expert who previously worked for NASA, has a great gamma FAQ that I recommend you read, especially if you plan to handle CGI. He also dispels several common misunderstandings concerning gamma.

- **LOGARITHMIC TRANSFORMATIONS**

Logarithmic Transformations (such as the Gamma Transformation, where gamma 1) can be used to brighten an image's intensity. It's most commonly used to boost the detail (or contrast) of low-intensity values. They're particularly good at bringing out detail in Fourier transformations (covered in a later lab). The equation for obtaining the Logarithmic transform of image f in MATLAB is:

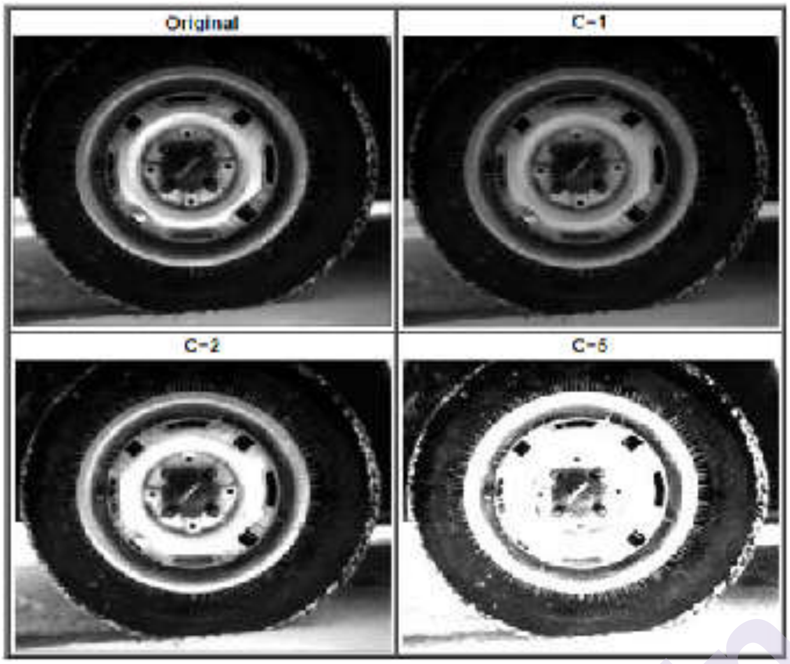
g = c*log(1 + double(f))

The constant c is typically used to scale the log function's range to fit the input domain. For an uint8 picture, c=255/log(1+255), or c=1/log(1+1) (~1.45) for a double image. It can also be used to boost contrast—the higher the c value, the brighter the image appears. The log function, when used in this manner, can produce results that are excessively bright to display. The graphic below shows the result for various values of c when a=0:.01:1. For the plots of c=2 and c=5, the min function clamps the y-values at 1. (teal and purple lines, respectively).






The original image and the outcomes of applying three of the transformations from above are shown below. When c=5, the image is the brightest, and the radial lines on the interior of the tyre can be seen (these lines are barely viewable in the original because there is not enough contrast in the lower intensities).

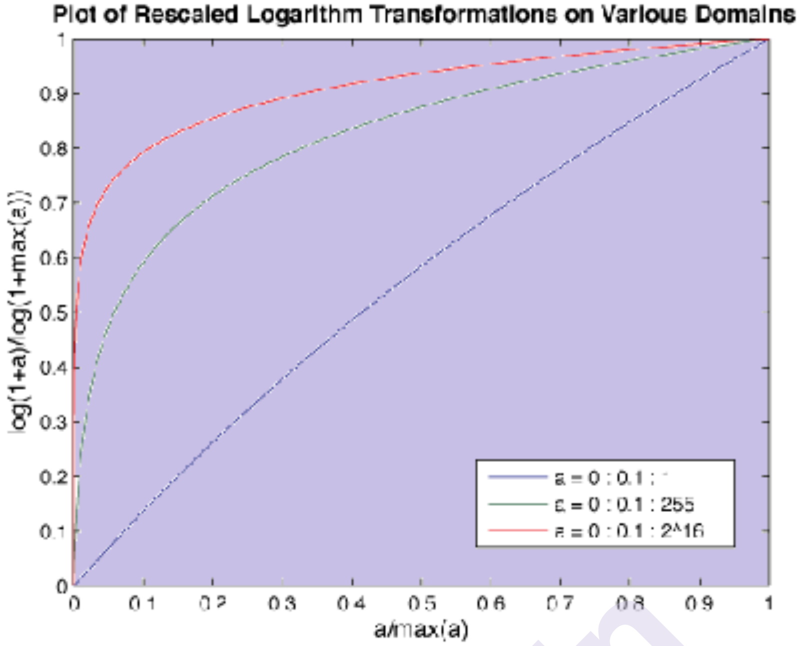



The MATLAB code that created these images is:

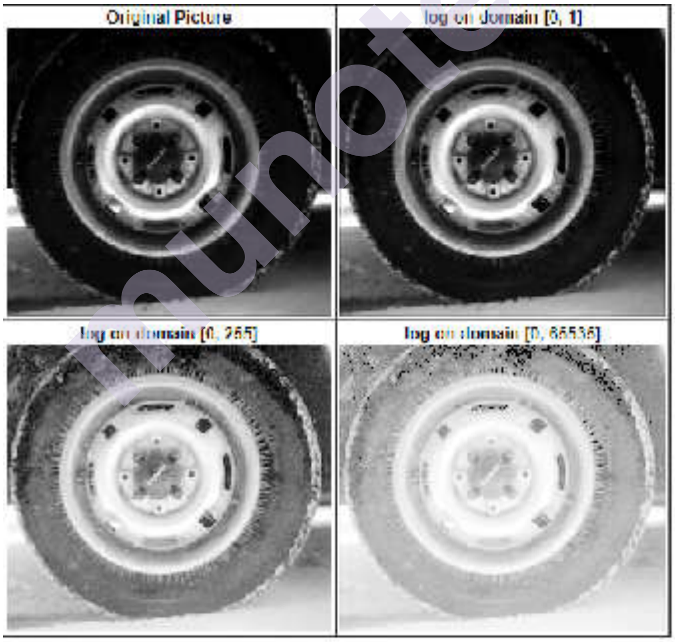
```
I=imread('tire.tif');
imshow(I)
I2=im2double(I);
J=1*log(1+I2);
J2=2*log(1+I2);
J3=5*log(1+I2);
figure, imshow(J)
figure, imshow(J2)
figure, imshow(J3)
```

Notice how the bright sections, when intensity levels are capped, lose detail. Any values generated by the scaling that is more than one are presented as 1 (full intensity) and should be capped. The min(matrix, upper bound) and max(matrix, lower bound) functions in MATLAB can be used to clamp data, as indicated in the legend for the plot above.

Although logarithms can be calculated in a variety of bases, including MATLAB's builtin log10, log2, and log (natural log), the resulting curve is the same for all bases when the range is scaled to match the domain. Instead, the curve's shape is determined by the range of values to which it is applied. Here are some log curve examples for a variety of input values:




If you want to use logarithm transformations properly, you should be aware of this effect. Here's what happens when you scale an image's values to those ranges before applying the logarithm transform:






The MATLAB code that produced these images is:

```
tire = imread('tire.tif');

d = im2double(tire);

figure, imshow(d);

%log on domain [0,1]
```

```
f = d;
c = 1/log(1+1);
j1 = c*log(1+f);
figure, imshow(j1);
%log on domain [0, 255]
f = d*255;
c = 1/log(1+255);
j2 = c*log(1+f);
figure, imshow(j2);
%log on domain [0, 2^16]
f = d*2^16;
c = 1/log(1+2^16);
j3 = c*log(1+f);
figure, imshow(j3);
```

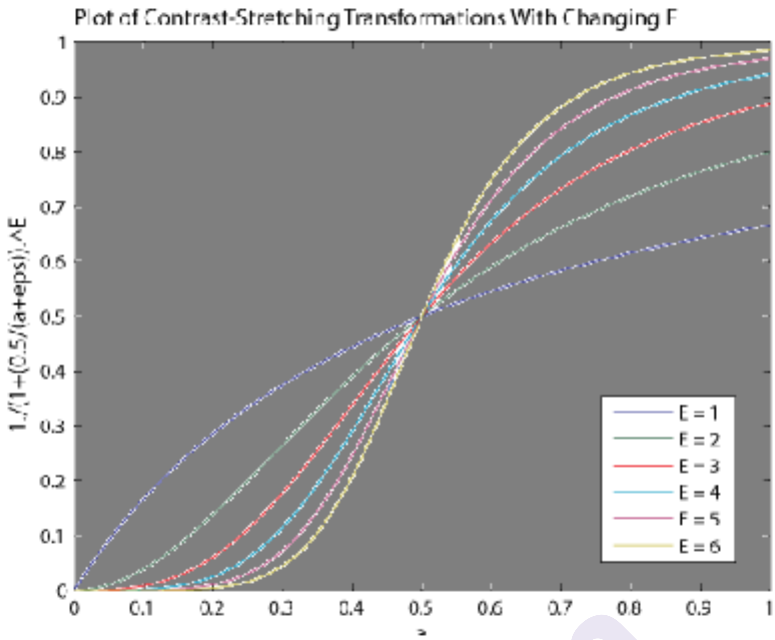
**The effects of the logarithm transform are barely evident in domain [0, 1], but they are greatly accentuated in domain [0, 65535]. It's also worth noting that, unlike linear scaling and clamping, gross detail remains visible in light areas.**

- **CONTRAST-STRETCHING TRANSFORMATIONS**

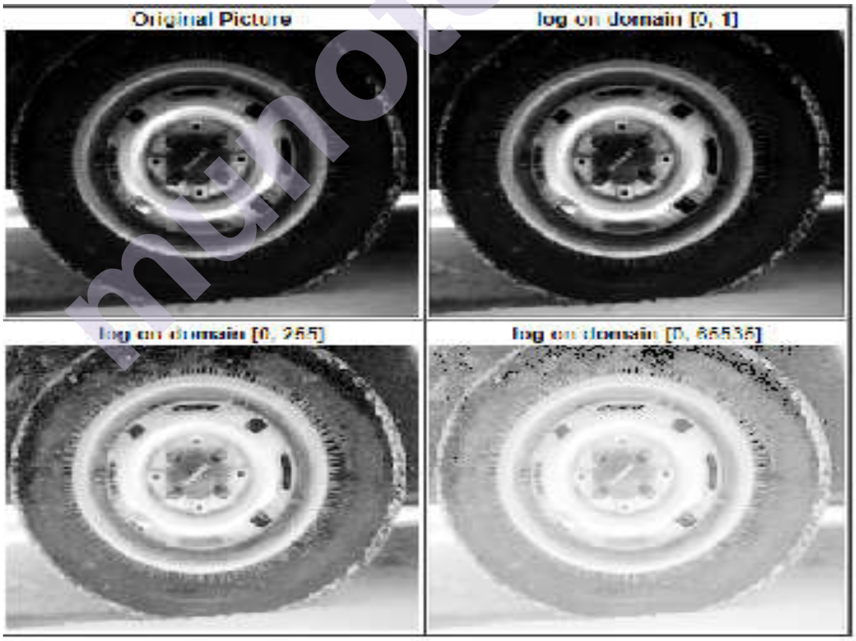
The contrast between the darks and the brightness is increased via contrast-stretching procedures. In lab 1, we saw a simplified version of section 5.3 of the textbook's automated contrast adjustment. That adjustment simply expanded the histogram to fill the image's intensity domain while keeping everything at about identical levels. You might want to push the intensity to a particular point every now and again. There are only a few degrees of grey around the level of interest, so everything darker darks are a lot darker and everything lighter is a lot lighter. In MATLAB, you can use the following function to make a contrast-stretching transformation:

**g=1./(1 + (m./(double(f) + eps)).^E)**

The function's slope is controlled by E, and the mid-point, m, is where you wish to switch from dark to bright values. The distance between 1.0 and the next greatest integer that may be expressed in a double-precision floating-point is represented by eps, a MATLAB constant. It is utilized in this equation to prevent division by zero if the image contains any zero-valued pixels. The outcomes of adjusting both m and E are represented in two plot/diagram sets below. Given a=0:.01:1 and m=0.5, the results for various values of E are plotted below.




The original image and the outcomes of applying the three changes from above are shown below. The m value used in the following examples is the average of the image intensities (0.2104). The function becomes more like a thresholding function with threshold m for very high E values,The resulting image is more black and white than grayscale, for example.

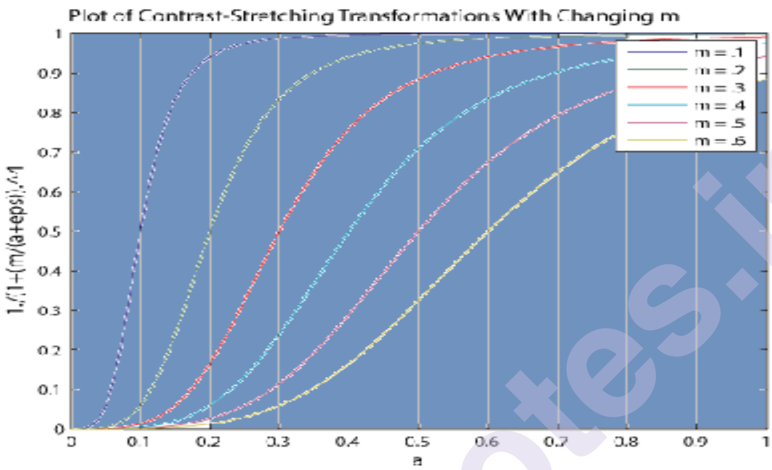



The MATLAB code that created these images is:

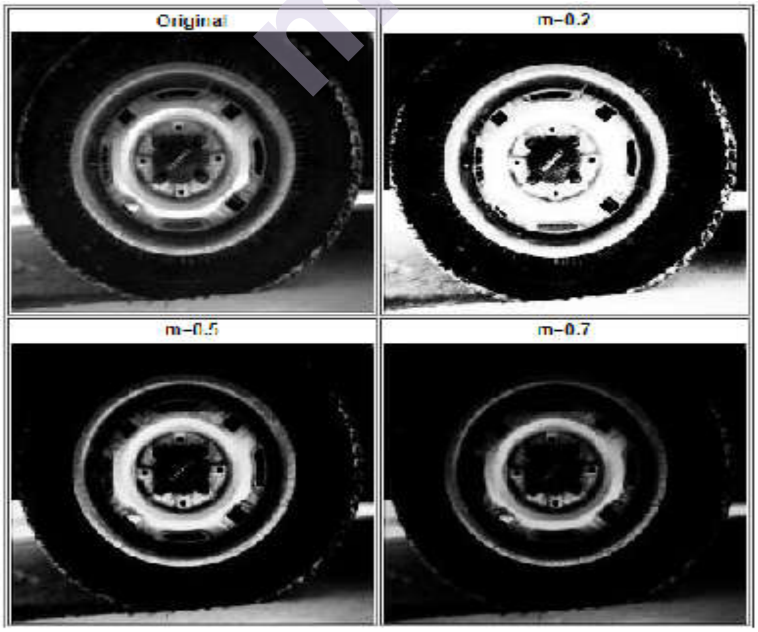
```
I=imread('tire.tif');

I2=im2double(I);

m=mean2(I2)

contrast1=1./(1+(m./(I2+eps)).^4);
```

```
contrast2=1./(1+(m./(I2+eps)).^5);

contrast3=1./(1+(m./(I2+eps)).^10);

imshow(I2)

figure,imshow(contrast1)

figure,imshow(contrast2)

figure,imshow(contrast3)
```

This second plot shows how changes to m (using E=4) affect the contrast curve:




The following shows the original image and the results of applying the three transformations from above. The m value used below is 0.2, 0.5, and 0.7. Notice that 0.7 produces a darker image with fewer details for this tire image.



The MATLAB code that created these images is:

```
I=imread('tire.tif');

I2=im2double(I);

contrast1=1./(1+(0.2./(I2+eps)).^4)

contrast2=1./(1+(0.5./(I2+eps)).^4);

contrast3=1./(1+(0.7./(I2+eps)).^4);

imshow(I2)

figure,imshow(contrast1)

figure,imshow(contrast2)

figure,imshow(contrast3)
```

- **The intrans and changeclass Functions**

Except for the contrast stretching transform, the file intrans.m Digital Image Processing, Using MATLAB[2] provides a function that performs all of the intensity transformations discussed above. You should go through the code and figure out how to implement that feature.

A second function named changeclass is used by the intrans function.

The intrans function's comments, which begin on the second line, explain how to use it. Please take note of the description of the missing contrast stretch transform, which states that it should take changing arguments and what defaults to use for missing values. The table below shows how intrans can be used to correlate to the four Intensity Transformation Functions. Consider the case when I=imread('tire.tif');

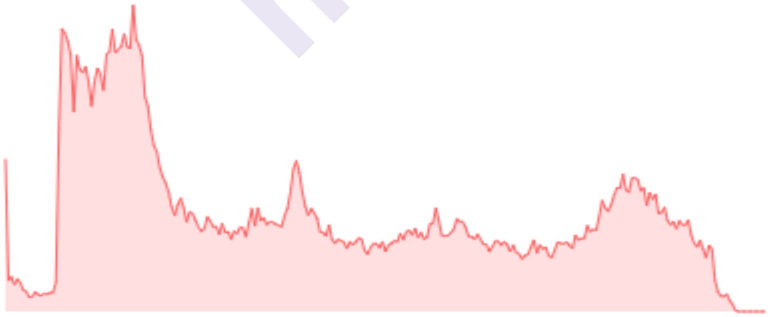
| **Transformation** | **Intensity Transformation Function** | **Corresponding intrans Call** |
|---|---|---|
| photographic negative | neg=imcomplement(I); | neg=intrans(I,'neg'); |
| logarithmic | I2=im2double(I); log=5*log(1+I2); | log=intrans(I,'log',5); |
| gamma | gamma=imadjust (I,[],[],0.4); | gamma=intrans(I,'gamma',0.4); |
| contrast-stretching | I2=im2double(I); contrast=1./(1+(0.2./(I2+eps)).^5); | contrast=intrans(I,'stretch',0.2,5); |

## 2.3.3 HISTOGRAM PROCESSING

- **HISTOGRAMS INTRODUCTION**

The histogram is a graphical representation of a digital image used in digital image processing. A graph is a representation of each tonal value as a number of pixels. In today's digital cameras, the image histogram is available. They are used by photographers to see the dispersion of tones captured.

The horizontal axis of a graph represents tonal fluctuations, whereas the vertical axis represents the number of pixels in that specific pixel. The left side of the horizontal axis depicts black and dark parts, the middle represents medium grey colour, and the vertical axis reflects the area's size.



**Histogram of the scenery**

***APPLICATIONS OF HISTOGRAMS***

1. Histograms are employed in software for simple computations in digital image processing.

2. It's a tool for analyzing images. A careful examination of the histogram can be used to predict image properties.

3. The image's brightness can be modified by looking at the histogram's features.

4. Having information on the x-axis of a histogram allows you to modify the image's contrast according to your needs.

5. It is used to equalize images. To create a high contrast image, the grey level intensities are extended along the x-axis.

6. Histograms are utilized in thresholding because they improve the image's appearance.

7. We can figure out which type of transformation is used in the method if we have the input and output histograms of an image.

*HISTOGRAM PROCESSING TECHNIQUES*

- **HISTOGRAM SLIDING**

The entire histogram is shifted rightwards or leftwards in histogram sliding. When a histogram is adjusted to the right or left, the brightness of the image changes dramatically. The intensity of light released by a particular light source determines the brightness of the image.

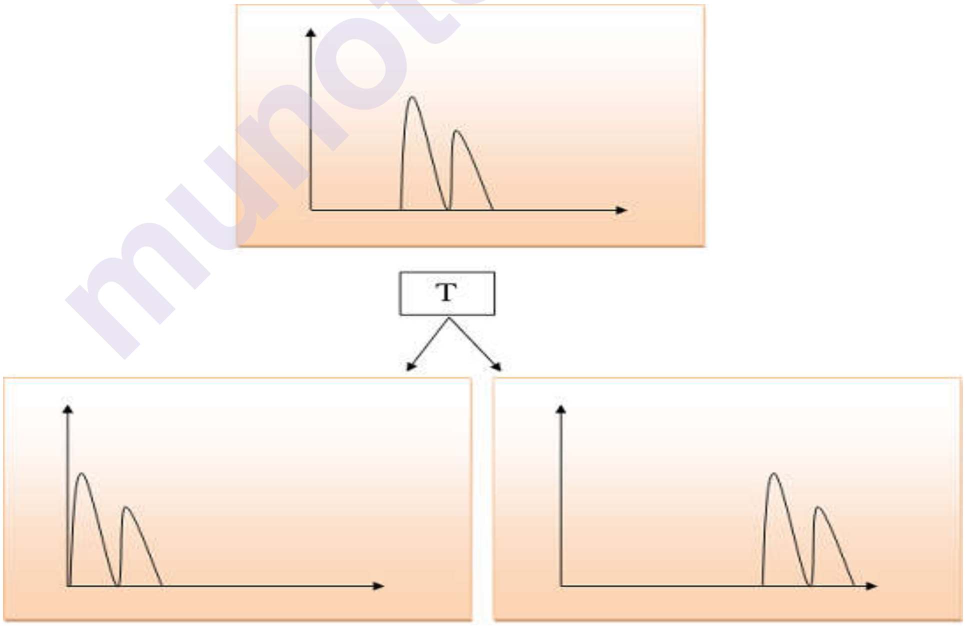


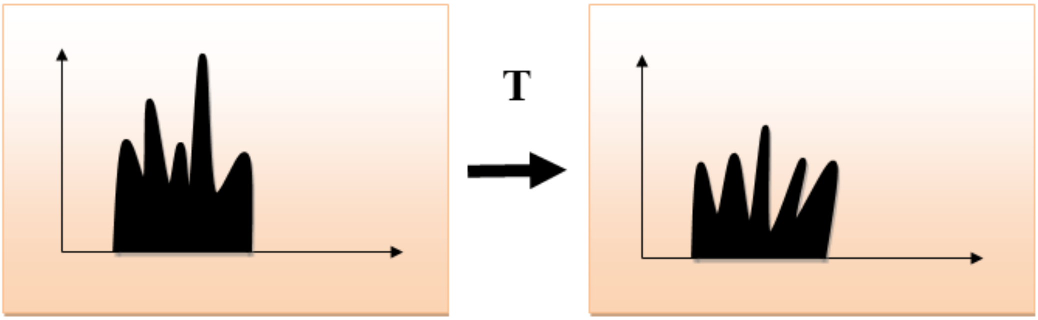

Fig. Histogram Sliding

- **HISTOGRAM STRETCHING**

The contrast of an image is boosted through histogram stretching. The contrast of an image is defined as the difference between the maximum and minimum pixel intensity values.

If we wish to increase the contrast of an image, we expand the histogram till it covers the entire dynamic range of the histogram.

We may determine whether an image has low or high contrast by looking at its histogram.

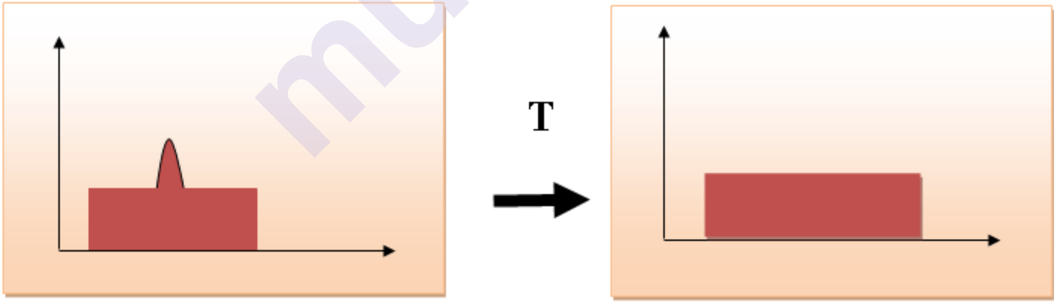



- **HISTOGRAM EQUALIZATION**

Equalizing all of an image's pixel values is done through histogram equalization. The transformation is carried out in such a way that the histogram is uniformly flattened.

Histogram equalization broadens the dynamic range of pixel values and ensures that each level has an equal number of pixels, resulting in a flat histogram with great contrast.

When extending a histogram, the shape of the histogram remains the same, however, when equalizing a histogram, the shape of the histogram changes, and just one image is generated.




## 2.3.4 IMAGE SUBTRACTION

- **IMAGE SUBTRACTION**

Image enhancement and segmentation (where an image is divided into various 'interesting' elements like edges and areas) are two applications for this approach. The foundations are built on subtracting two images, which is defined as computing the difference between each pair of related pixels in the two images. It can be written as:

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

A fascinating application is in medicine, where h(x,y) is called a mask and subtracted from a succession of photos fi(x,y), yielding some fascinating images. It is possible to watch a dye propagate through a person's brain arteries, for example, by doing so. The portions in the photos that look the same get darkened each time the difference is calculated, while the differences become more highlighted (they are not subtracted out of the resulting image).

- **IMAGE AVERAGING**

Consider a noisy image g(x,y), which is created by adding a specific amount of noise n(x,y) to an original image f(x,y):

$$g(x, y) = f(x, y) + \eta(x, y)$$

The noise is expected to be uncorrelated (thus homogeneous across the image) and have an average value of zero at each pair (x,y). By introducing a set of noisy images g(x,y), the goal is to lessen the noise effects.

Assume we have an image that was created by averaging noisy images:

$$\overline{g}(x, y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x, y).$$

We now calculate the expected value of $\overline{g}$ which is :

$$\begin{aligned} E\{\overline{g}(x, y)\} &= E\left\{\frac{1}{M}\sum_{i=1}^{M} g_i(x, y)\right\} \\ &= \frac{1}{M}E\left\{\sum_{i=1}^{M} g_i(x, y)\right\} \\ &= \frac{1}{M}\sum_{i=1}^{M} E\{g_i(x, y)\} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{M}\sum_{i=1}^{M} E\{f(x,y)+\eta_i(x,y)\} \\
&= \frac{1}{M}\sum_{i=1}^{M}(E\{f(x,y)\}+E\{\eta_i(x,y)\}) \\
&= \frac{1}{M}\sum_{i=1}^{M}(f(x,y)+0) \\
&= \frac{1}{M}Mf(x,y) \\
&= f(x,y).
\end{aligned}
$$

## 2.4 LET US SUM UP

Enhancement aims to improve the quality of an image so that it may be used in a certain process. The word spatial refers to the Image Plane itself, which is DIRECT pixel manipulation. Frequency domain processing approaches work by altering an image's Fourier transform. Equalization of histograms is a typical approach for improving the appearance of photographs. We need to identify a transformation T that converts grey values r in the input image F to grey values s = T(r) in the converted image.

Figure 2 shows the original image, histogram, and equalized versions. Image smoothing can be done in a variety of ways. We'll look at edge-preserving smoothing. The average pixel value is obtained from the average pixel value in a neighborhood of (x,y) in the input image. Other mask shapes can cause strange things to happen to the image's frequency spectrum.

## 2.5 LIST OF REFERENCES


1. https://www.mv.helsinki.fi/home/khoramsh/4Image%20Enhancement%20in%20Spatial%20Domain.pdf

2. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node3.html

3. https://www.sciencedirect.com/topics/engineering/spatial-domain

4. http://www.faadooengineers.com/online-study/post/cse/digital-imge-processing/674/spatial-domain-methods

5. **https://www.google.com/search?q=point+processing+in+image+processing&rlz=1C1CHZN_enIN974IN974&oq=POINT+PROCESSING&aqs=chrome.1.0i512l10.1767j0j15&sourceid=chrome&ie=UTF-8**

6. https://www.cs.uregina.ca/Links/class-info/425/Lab3/

7. https://www.javatpoint.com/dip-histograms#:~:text=In%20digital%20image%20processing%2C%20histograms,the%20details%20of%20its%20histogram.

8. **http://www.faadooengineers.com/online-study/post/ece/digital-image-processing/1123/image-subtraction-and-image-averaging**


## 2.6 BIBLIOGRAPHY


1. https://www.mv.helsinki.fi/home/khoramsh/4Image%20Enhancement%20in%20Spatial%20Domain.pdf
2. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node3.html
3. https://www.sciencedirect.com/topics/engineering/spatial-domain
4. http://www.faadooengineers.com/online-study/post/cse/digital-imge-processing/674/spatial-domain-methods
5. **https://www.google.com/search?q=point+processing+in+image+processing&rlz=1C1CHZN_enIN974IN974&oq=POINT+PROCESSING&aqs=chrome.1.0i512l10.1767j0j15&sourceid=chrome&ie=UTF-8**
6. https://www.cs.uregina.ca/Links/class-info/425/Lab3/
7. https://www.javatpoint.com/dip-histograms#:~:text=In%20digital%20image%20processing%2C%20histograms,the%20details%20of%20its%20histogram.
8. **http://www.faadooengineers.com/online-study/post/ece/digital-image-processing/1123/image-subtraction-and-image-averaging**


## 2.7 UNIT END EXERCISES

1. What is the goal of spatial domain image enhancement?
2. What are the different types of filters used in the spatial domain?
3. What Did You Mean When You Said "Digital Image Shrinking"?
4. What are intensity transformations and how do they work?
5. Which of the following processes broadens the range of intensity levels?
6. In digital image processing, what is histogram processing?
7. What exactly is the point of image subtraction?
8. How does applying an average filter to a digital image affect it?
9. What are the most common applications for smoothing filters?
10. Why is frequency domain preferable to time domain?

❖❖❖❖

# 3

# IMAGE AVERAGING SPATIAL FILTERING

**Unit Structure**





## 3.0 OBJECTIVES

- The Spatial Filtering technique is applied to individual pixels in an image. A mask is typically thought to be increased in size so that it has a distinct center pixel. This mask is positioned on the image so that the mask's center traverses all of the image's pixels.

- Spatial filtering is frequently used to "clean up" laser output, reducing aberrations in the beam caused by poor, unclean, or damaged optics, or fluctuations in the laser gain medium itself.

## 3.1 INTRODUCTION

Spatial filtering is a method of modifying the features of an optical image by selecting deleting certain spatial frequencies that make up an object, such as video data received from satellites and space probes, or raster removal from a television broadcast or scanned image.

Average (or mean) filtering is a technique for smoothing photographs by lowering the intensity fluctuation between adjacent pixels. The average filter replaces each value with the average value of neighboring pixels, including itself, as it moves through the image pixel by pixel.

*Filtering is a method of altering or improving an image. The processed value for the current pixel depends on both itself and adjacent pixels in a spatial domain operation or filtering... Filters or masks will be defined.*

Filtering is a method of altering or improving an image. The processed value for the current pixel depends on both itself and adjacent pixels in a spatial domain operation or filtering... Filters or masks will be defined.

## 3.2 AN OVERVIEW

### IMAGE ENHANCEMENT OVERVIEW

**B**y working with noisy photos we can filter signals from noise in two dimensions. Two types of noise: binary and Gaussian.

The user specifies a percentage value in the binary case (a number between 0 and 100). This value is randomly set equal to the maximum grey level value and reflects the percentage of pixels in the image whose values will be completely lost (corresponding to a white pixel).

The value of the pixel x(k,l) is changed in the Gaussian case by additive white gaussian noise x(k,l)+n, with noise n~N(0,v) being normally distributed and variance v set by the user (a number between 0 and 2 in this exercise).

The image is the same in binary noise, except for a set of points where the image's pixels are set to white. The noisy image seems blurred in the case of Gaussian noise.

**Original Image**

**Image with binary noise**

**Image with Gaussian noise**

The method of removing noise or sharpening photographs to increase image quality is known as image enhancement. Even though image enhancement is a well-established approach, we will concentrate on two strategies based on the notion of filtering an original image to produce a restored or better image. Both linear and nonlinear actions are possible with our filters.

## 1. Median filtering

A pixel is replaced by the median of the pixels in a window around it in median filtering. That is to say,

W is a suitable window that surrounds the pixel. The median filtering algorithm entails sorting the pixel values in the window in ascending or descending order and selecting the middle value. In most cases, a square window with an odd square size is chosen.

## 2. Spatial averaging

Each pixel is replaced by an average of its nearby pixels in the case of spatial averaging. That is to say,

$$y(m,n) = \frac{1}{N_w} \sum\sum_{(k,l) \in W} x(m-k, n-l)$$

Where W is the number of pixels in the window, and Nw is the number of pixels in the window. Because spatial averaging causes a distortion in the form of blurring, the size of the window W is limited in practice.

To introduce noise to an image and then recover it using the techniques as described above. You'll notice that the best picture enhancement strategy is determined by the type of noise as well as the amount and level of noise in the image.

## 3.3 IMAGE AVERAGING AND SPATIAL FILTERING

**SPATIAL FILTERING AND ITS TYPES**

The Spatial Filtering technique is applied to individual pixels in an image. A mask is typically thought to be increased in size so that it has a distinct centre pixel. This mask is positioned on the image so that the mask's centre traverses all of the image's pixels.

**Using linearity as a criterion for classification:**

**There are two kinds of them:**

**1.** Linear Spatial Filter

**2.** Non-linear Spatial Filter

**Classification in General:**

Smoothing Spatial Filter: A smoothing spatial filter is used to blur and reduce noise in an image. Blurring is a pre-processing technique for removing minor details, and it is used to achieve Noise Reduction.

**Types of Smoothing Spatial Filter:**

**1.** Linear Filter (Mean Filter)

**2.** Order Statistics (Non-linear) filter

These are explained in the next paragraphs.

1. Mean Filter: A linear spatial filter is just the average of the pixels in the filter mask's neighborhood. The goal is to replace the value of each pixel in a picture with the average of the grey levels in the filter mask's neighborhood.

**Types of Mean filter:**

(i) Averaging filter: This filter is used to reduce image detail. The coefficients are all the same.

(ii) Weighted averaging filter: Pixels are multiplied by various coefficients in this filter. The average filter is multiplied by a higher value than the centre pixel.

**1. Order Statistics Filter:**

This filter is based on the ordering of pixels within the image region it covers. It substitutes the value indicated by the ranking result for the value of the centre pixel. This filtering preserves the edges better.

(i) **Minimum filter:** The 0th percentile filter is the smallest of the order statistics filters. The smallest value in the window replaces the value in the center.

**(ii) Maximum filter**: The maximum filter is the 100th percentile filter. The largest value in the window replaces the value in the center.

**(ii) Median filter**: Every pixel in the image is taken into account. The original values of the pixel are replaced by the median of the list after surrounding pixels are sorted first.

**Sharpening Spatial Filter :**

(also known as a derivative filter) is a type of spatial filter that sharpens the image. The sharpening spatial filter serves the exact opposite objective as the smoothing spatial filter. Its primary goal is to eliminate blurring and highlight the edges. The first and second-order derivatives are used.

**First order derivative:**

- Must be zero in flat segments.
- Must be non zero at the onset of a grey level step.
- Must be non zero along ramps.

First order derivative in 1-D is given by:

$f' = f(x+1) - f(x)$

**Second order derivative:**

- Must be zero in flat areas.
- Must be zero at the onset and end of a ramp.
- Must be zero along ramps.

Second order derivative in 1-D is given by:

$f'' = f(x+1) + f(x-1) - 2f(x)$

## 3.3.1 SMOOTHING FILTERS

**SMOOTHING FILTERS**

To reduce the amount of noise in an image, image smoothing filters such as the Gaussian, Maximum, Mean, Median, Minimum, Non-Local Means, Percentile, and Rank filters can be used. Although these filters can efficiently reduce noise, they must be applied with caution so that crucial information in the image is not altered. It's also worth noting that, in most circumstances, edge detection or augmentation should come after smoothing.

- **GAUSSIAN**
- **MEAN**
- **MEAN SHIFT**

- **MEDIAN**
- **NON-LOCAL MEANS**
- **GAUSSIAN**

When you apply the Gaussian filter to an image, it blurs it and removes information and noise. It's comparable to the Mean filter in this regard. It does, however, use a kernel that represents a Gaussian or bell-shaped hump. Unlike the Mean filter, which produces an evenly weighted average, the Gaussian filter produces a weighted average of each pixel's neighborhood, with the average weighted more towards the center pixels' value. As a result, the Gaussian filter smoothes the image more gently and maintains the edges better than a Mean filter of comparable size.

The frequency response of the Gaussian filter is one of the main justifications for adopting it for smoothing. Lowpass frequency filters are used by the majority of convolution-based smoothing filters. As a result, they have the effect of removing high spatial frequency components from an image. You can be quite certain about what range of spatial frequencies will be present in the image after filtering by selecting an adequately big Gaussian, which is not the case with the Mean filter. Computational biologists are also interested in the Gaussian filter since it has been associated with some biological plausibility. For example, some cells in the brain's visual circuits often respond in a Gaussian fashion.

Because many edge-detection filters are susceptible to noise, Gaussian smoothing is typically utilised before edge detection.

**MEAN**

Mean filtering is a straightforward technique for smoothing and reducing noise in photographs by removing pixel values that aren't indicative of their surroundings. Mean filtering is a technique that replaces each pixel value in an image with the mean or average of its neighbors, including itself.

The Mean filter, like other convolution filters, is based on a kernel, which describes the shape and size of the sampled neighborhood for calculating the mean. The most common kernel size is 3x3, but larger kernels might be utilized for more severe smoothing. It's worth noting that a small kernel can be applied multiple times to achieve a similar, but not identical, result to a single pass with a large kernel.

Although noise is reduced after mean filtering, the image has been softened or blurred, and high-frequency detail has been lost. This is mainly caused by the filter's limits, which are as follows:

• A single pixel with a very atypical value can have a considerable impact on the mean value of all the pixels in its vicinity.

The filter will interpolate new values for pixels on the edge when the filter neighborhood straddles an edge. If crisp edges are required in the output, this could be a problem.

The Median filter, which is more commonly employed for noise reduction than the Mean filter, can solve both of these concerns. Smoothing is often done with other convolution filters that do not calculate the mean of a neighborhood. The Gaussian filter is one of the most popular.

**MEAN SHIFT**

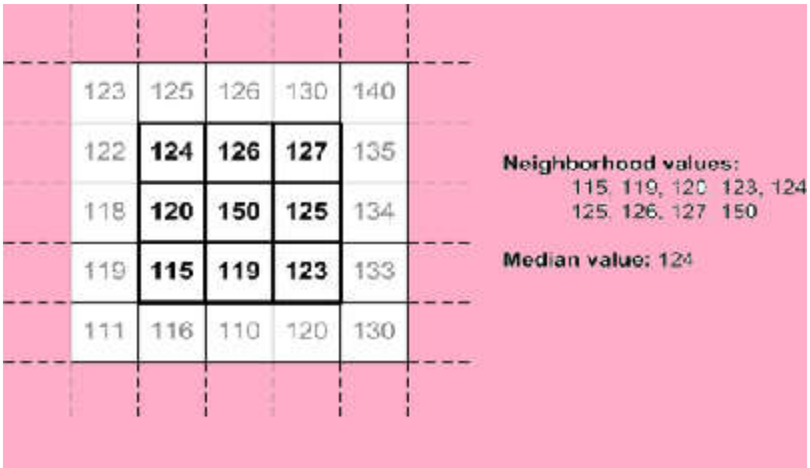
Mean shift filtering is based on a data clustering algorithm extensively used in image processing and can be utilized for edge-preserving smoothing. The collection of surrounding pixels is determined for each pixel in an image with a spatial location and a specific grayscale value. The new spatial center (spatial mean) and the new mean value are calculated for this set of adjacent pixels. The new center for the following iteration is determined by the calculated mean values. Iterate the specified technique until the spatial and grayscale mean cease changing. The final mean value will be set to the iteration's beginning point at the end of the iteration.

**MEDIAN**

The Median filter is typically used to minimise image noise, and it can often preserve image clarity and edges better than the Mean filter. This filter, like the Mean filter, examines each pixel in the image individually and compares it to its neighbours to determine whether it is typical of its surroundings. Instead of merely replacing the pixel value with the mean of nearby pixel values, the median of those values is used instead. Median filters are especially good for reducing random intensity spikes that commonly appear in microscope images.

This filter's operation is depicted in the diagram below. The median is derived by numerically ordering all of the pixel values in the surrounding neighborhood, in this case, a 3x3 square, and then replacing the pixel in question with the middle pixel value.

**Median filter**



The center pixel value of 150, as seen in the picture, is not typical of the surrounding pixels and is substituted with the median value of 124. It's worth noting that larger neighborhoods will result in more severe smoothing.

The Median filter has two key advantages over the Mean filter since it calculates the median value of a neighborhood rather than the mean: • The median is more robust than the mean, thus a single very unrepresentative pixel in a neighborhood will not have a substantial impact on the median value. For example, in datasets contaminated with salt-and-pepper noise (scatter dots).

- Since the median value must be the value of one of the pixels in the neighborhood, the Median filter does not create unrealistic pixel values when the filter straddles an edge. For this reason, it is much better at preserving sharp edges than the Mean filter.
- However, the Median filter is sometimes not as subjectively good at dealing with large amounts of Gaussian noise as the Mean filter. It is also relatively complex to compute.

**NON-LOCAL MEANS**

Unlike the Mean filter, which smooths a picture by taking the mean of a set of pixels surrounding a target pixel, the Non-Local Means filter takes the mean of all pixels in the image, weighted by their similarity to the target pixel. When compared to mean filtering, this filter can result in improved post-filtering clarity with minimum information loss. When smoothing noisy images, the Non-Local Means or Bilateral filter should be your first choice in many circumstances.

It's worth noting that non-local means filtering works best when the noise in the data is white noise, in which case most visual characteristics, including small and thin ones, will be maintained.

## 3.3.2 SHARPENING FILTERS

Image preprocessing has long been a feature of computer vision, and it can considerably improve the performance of machine learning models. Image processing is the process of applying several sorts of filters to our image. Filters can assist minimize image noise while also enhancing the image's qualities.

**Sharpening filters are discussed as below.**

*• When compared to smooth and blurry images, sharpening filters make the transition between features more recognizable and evident.*

*• What occurs when a sharpening filter is applied to an image?*

*When compared to their neighbors, the brighter pixels are rendered brighter (boosted).*

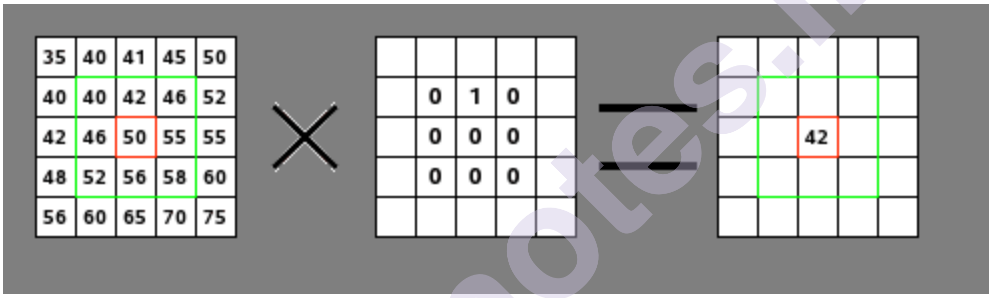
*Sharpening or blurring an image can be reduced to a series of matrix arithmetic operations.*

*When we apply a filter to our image, we're doing a convolution operation on it with a Xen kernel. A kernel is a square matrix with nxn dimensions.*

***CONVOLUTION AND KERNEL***

Each image can be represented as a matrix, with its features represented as numerical values, and we use convolution with various types of matrices known as kernels to extract or enhance distinct features.

The act of adding each element of the image to its nearby neighbors, weighted by the kernel, is known as convolution. This has something to do with a type of mathematical convolution. Despite being marked by "*," the matrix operation being performed—convolution—is not ordinary matrix multiplication.




The kernel is what determines the type of operation we're doing, such as sharpening, blurring, edge detection, gaussian blurring, and so on.

The following is an example of a sharpening kernel:

| | |
|---|---|
| **Sharpen** | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |

**SHARPENING**

• Sharpening is a technique for enhancing the transition between features and details by sharpening and highlighting the edges. Sharpening, on the other hand, does not consider whether it is enhancing the image's original features or the noise associated with it. It improves both.

**Blurring vs Sharpening**

- Blurring: Blurring/smoothing is accomplished in the spatial domain by averaging the pixels of its neighbors, resulting in a blurring effect. It's an integration procedure.

- Sharpening: Sharpening is a technique for identifying and emphasizing differences in the neighborhood. It is a differentiation process.

Sharpening Filters of Various Types

1) High Boost Filtering and Unsharp Masking

Using a smoothing filter, we can sharpen an image or perform edge improvement.

1. Make the image blurry. Blurring is the process of suppressing the majority of high-frequency components.

2. Original Image - Blurred Image (Output (Mask)). Most of the high-frequency components that were previously blocked by the blurring filter are now present in this output.

3. By applying the mask to the original image, the high-frequency components will be enhanced.

This procedure is called UNSHARP MASKING since we are using a blurred image to create our personalized mask.

As a result, Unsharp Mask m(x, y) can be written as:

$$m(x,y) = f(x,y) - f_b(x,y)$$

- f(x,y) = original image.
- fb(x,y) = blurred image.

When you apply this mask to the original image, the high frequency components are enhanced.

$$g(x,y) = f(x,y) + k * m(x,y)$$

The value k determines how much weight should be given to the mask that is being added.

1. Unsharp Masking is represented by k = 1.

2. High Boost Filtering is represented by k > 1 since we are boosting high-frequency components by adding higher weights to the image's mask (edge features).

This approach, like most other sharpening filters, will not yield adequate results if the image contains noise.

We may get the mask without subtracting the blurred image from the original by using a negative Laplacian filter.

### *2) Laplacian Filters*

A second-order derivative mask is a Laplacian Filter. It attempts to eliminate the INWARD and OUTWARD edges. This difference in second-order derivatives aids in determining whether the changes we're seeing are caused by pixel changes in continuous regions or by an edge.

Positive values are found at the center of a general Laplacian kernel, while negative values are found in a cross pattern.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

To proceed with the derivation of this kernel matrix, knowledge of partial derivatives and Laplacian operators is required.

Let us consider our image as function of two variables, **f(x, y)**. We will be dealing with partial derivatices along the two spatial axes.

Gradient operator (linear operator) $$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator (non-linear) $$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

**Discrete form of Laplacian**

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

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

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

**Resultant resultant Laplacian Matrix**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | –4 | 1 |
| 0 | 1 | 0 |

***Laplacian Operators' Effects***

*• It emphasizes and intensifies grey discontinuities while deemphasizing continuous regions (regions without edges), i.e. derivatives that vary slowly.*

*We'll utilize some approximate Laplacian Filters for our programming.*

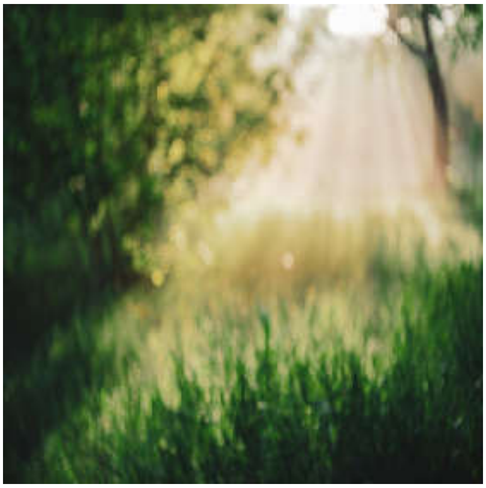
*Let us perform sharpening using different methods*

Using OpenCV as a tool

OpenCV is a python-based library for dealing with computer vision issues.

Let's have a look at the code below and figure out what's going on.

- We'll start by importing the libraries we'll need to sharpen our image.
- Numpy -> For conducting quick matrix operations OpenCV -> For image operations
- cv2.imread -> cv2.imread -> cv2.imread -> To read the input image from our disc in the form of a numpy array.
- cv2.scale -> To resize our image to fit in the dimensions of (400, 400).



• kernel -> kernel is a 3X3 matrix that we define based on how we want to slide the picture across for convolution.

• cv2.filter2D -> cv2.filter2D -> cv2.filter2D To convolve a kernel with an image, Opencv includes a function called filter2D.

It accepts three parameters as input:

1. img -> picture input

2. ddepth -> the depth of the output image

3. kernel-> kernel of convolution

This is how we can use OpenCV to conduct sharpening.

Changing the magnitudes of the kernel matrix allows us to experiment with the kernel to obtain different levels of sharpened images.

## Original Image

• ImageFilter has a number of pre-defined filters, such as sharpen and blur, that may be used with the filter() method.

• We sharpen our image twice and save the results in the sharp1 and sharp2 variables.

**Image after 1st sharp operation**

Image after 2nd sharp operation

Sharpening effects can be seen, with the features becoming brighter and more distinguishable.

## 3.4 FREQUENCY DOMAIN METHOD

***Frequency domain methods***

*In the frequency domain, image enhancement is simple. To create the enhanced image, we simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter (rather than convolve in the spatial domain), and then take the inverse transform.*

*The concept of blurring an image by lowering the magnitude of its high-frequency components or sharpening an image by increasing the amplitude of its high-frequency components is intuitively simple. However, implementing similar actions as convolutions by modest spatial filters in the spatial domain is typically more computationally efficient. Understanding frequency domain principles is crucial since it leads to enhancement approaches that would otherwise go unnoticed if attention was focused solely on the spatial domain.*

**Filtering**

Low pass filtering is the process of removing high-frequency components from an image. The image is blurred as a result of this (and thus a reduction in sharp transitions associated with noise). All low-frequency components would be retained while all high-frequency components would be eliminated in an ideal low pass filter. Ideal filters, on the other hand, have two flaws: blurring and ringing. The shape of the related spatial domain filter, which includes a huge number of undulations, is the source of these issues. Smoother frequency-domain filter transitions, such as the Butterworth filter, produce substantially superior outcomes.

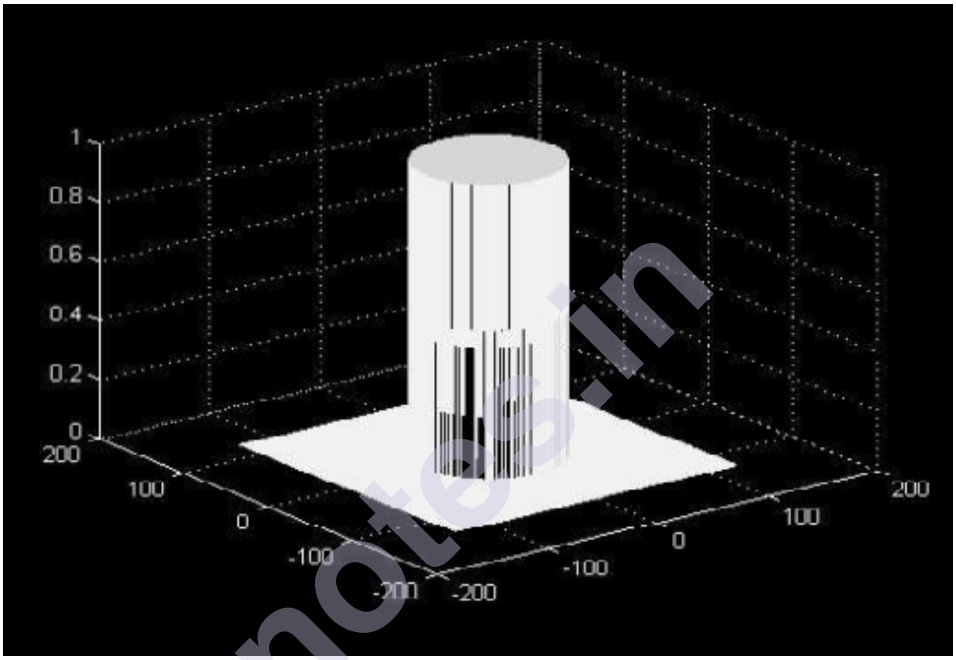
Figure 5: An ideal low pass filter's transfer function.




## 3.4.1 LOW PASS FILTERING

The high-frequency content of an image's Fourier transform is heavily influenced by edges and sudden changes in gray values. • In an image, regions of relatively uniform gray values contribute to the Fourier transform's low-frequency content. • As a result, a picture can be smoothed in the Frequency domain by lowering the Fourier transform's high-frequency content. This is a lowpass filter, right? • For the sake of simplicity, we'll just discuss real and radially symmetric filters. • A perfect lowpass filter with r0 as the cutoff frequency

$$H(u,v) = \begin{cases} 1, \text{ if } \sqrt{u^2+v^2} \le r_0 \\ 0, \text{ if } \sqrt{u^2+v^2} > r_0 \end{cases}$$




**Ideal LPF with r0 = 57**

The origin (0, 0) is in the image's center, not its corner (remember the "fftshift" operation).

• Using electrical components, the sudden shift from 1 to 0 of the transfer function H (u,v) is impossible to achieve in practice. It can, however, be simulated on a computer.

**Ideal LPF examples**

Original Image

LPF image, $r_0 = 57$

LPF image, $r_0 = 36$

LPF image, $r_0 = 26$

The blurred images have a pronounced ringing effect, which is a hallmark of perfect filters. The discontinuity in the filter transfer function is to blame.

In an ideal LPF, the cutoff frequency is chosen.

• The number of frequency components passed by the filter is determined by the ideal LPF's cutoff frequency 0 r.

• The smaller the 0 r value, the more image components are removed by the filter.

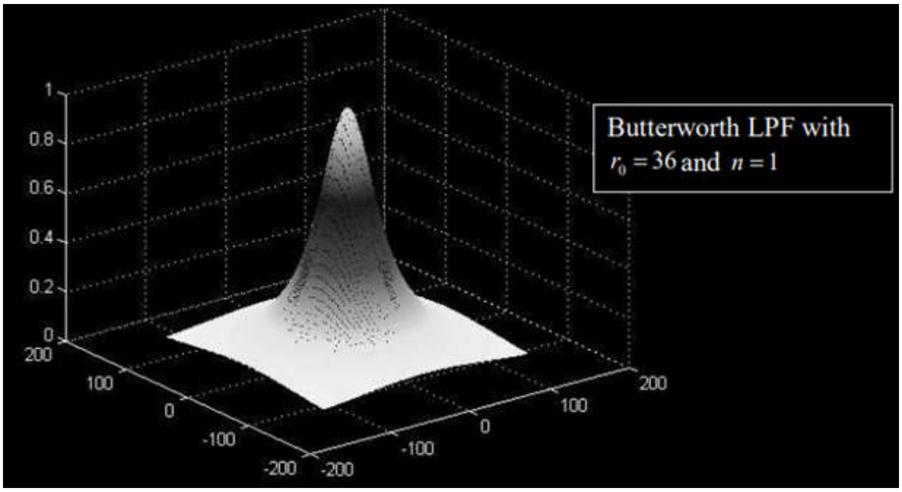
• In general, the value of 0 r is selected so that the majority of the components of interest pass through while the majority of the non-interesting components are deleted. This is usually a set of contradictory needs. We'll look at some of the specifics of image restoration.

• Computing circles that contain a given fraction of the total picture power is a good technique to establish a set of standard cut-off frequencies.

• Suppose − = − = = 1 0 1 0 ( , ) N v M u TP P u v , where 2 P(u,v) = F(u,v) , is the total image power.

• Consider a circle of radius () r0 α as a cutoff frequency in relation to a threshold, such that T v u ∑∑P(u,v) = αP

• After that, we can set a threshold and calculate an acceptable cutoff frequency () r0 α.




• A two-dimensional Butterworth lowpass filter has the following transfer function:

• r0: cutoff frequency, n: filter order

• Because the frequency response does not have a fast transition like the ideal LPF, it is better for image smoothing because it does not introduce ringing. n r u v H u v 2 0

Butterworth LPF example

Original Image

LPF image, r0 =18

LPF image, $r_0 = 13$

LPF image, $r_0 = 10$

**Butterworth LPF example: False contouring**

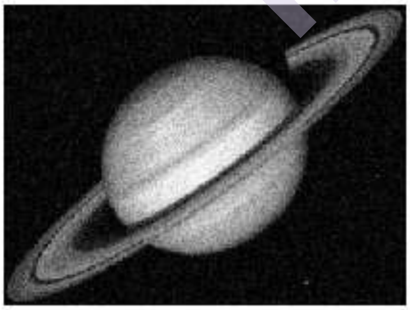
**Image with false contouring due to insufficient** **Lowpass filtered version of previous image**

**bits used for quantization**

Butterworth LPF example: Noise filtering

Butterworth LPF example: Noise filtering

Original Image



Noisy Image

LPF Image

Low-pass Gaussian filters

• In two dimensions, the form of a Gaussian lowpass filter is 2 2 ( , ) / 2 ( , ) − σ = D u v H u v e , where 2 2 D(u,v) = u + v is the frequency plane distance from the origin.

• The parameter σ represents the Gaussian curve's spread or dispersion. The greater the value of σ , the higher the cutoff frequency and the less severe the filtering.

• The filter is reduced to 0.607 of its maximum value of 1 when D(u,v) =σ

## 3.4.2 HIGH PASS FILTERING

**HIGHPASS FILTERING**

• The high-frequency content of a Fourier transform is heavily influenced by edges and sudden transitions in gray values in a picture.

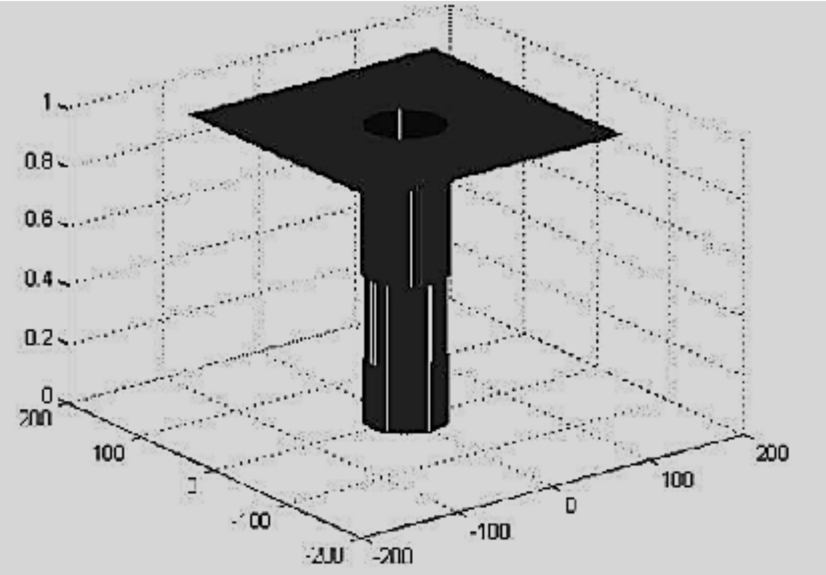
• Low-frequency content of a Fourier transform is influenced by regions of relatively uniform gray values in an image.

• As a result, image sharpening in the Frequency domain can be accomplished by lowering the Fourier transform's low-frequency content. A highpass filter would be this.

• Only real and radially symmetric filters will be considered for the sake of simplicity.

• With a cutoff frequency of 0 r, an ideal highpass filter is:

$$H(u,v)=\begin{cases}0, \text{ if } \sqrt{u^2+v^2}\le r_0\\ 1, \text{ if } \sqrt{u^2+v^2}>r_0\end{cases}$$



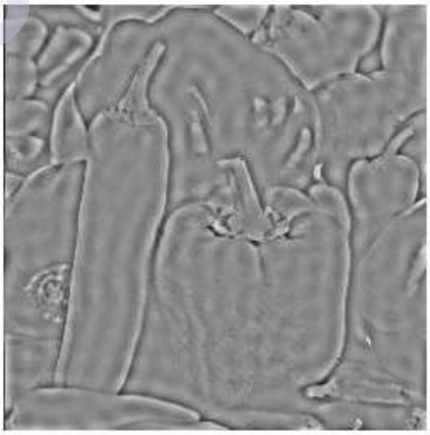

**Ideal HPF with r0 = 36**

The origin (0, 0) is in the image's centre, not its corner (remember the "fftshift" operation).

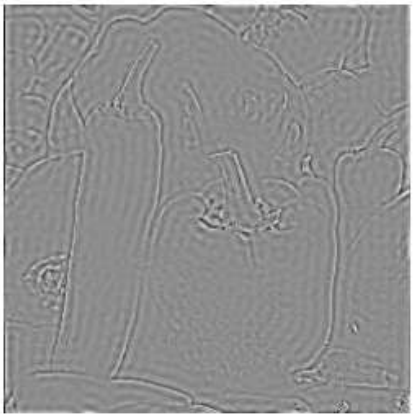
• Using electrical components, the sudden shift from 1 to 0 of the transfer function H (u,v) is impossible to achieve in practise. It can, however, be simulated on a computer.

## Ideal HPF examples

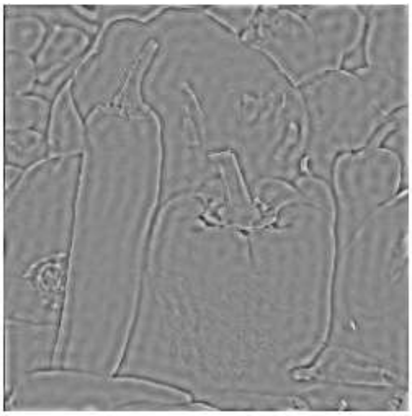
Original Image



HPF image, $r_0 = 18$



HPF image, $r_0 = 36$



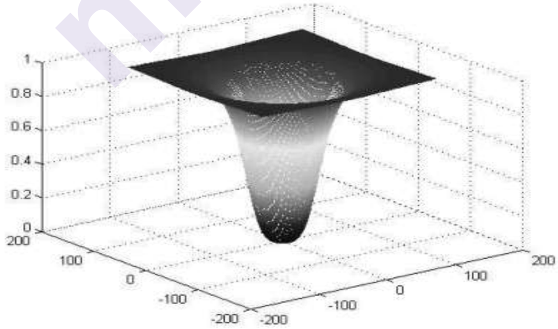
HPF image, $r_0 = 26$

- Note how the output images have a strong ringing effect, which is a hallmark of ideal filters. The discontinuity in the filter transfer function is to blame.

- A two-dimensional Butterworth highpass filter has the following transfer function:

$$H(u,v) = \frac{1}{1 + \left[\dfrac{r_0}{\sqrt{u^2 + v^2}}\right]^{2n}}$$

- n: filter order, r0: cutoff frequency

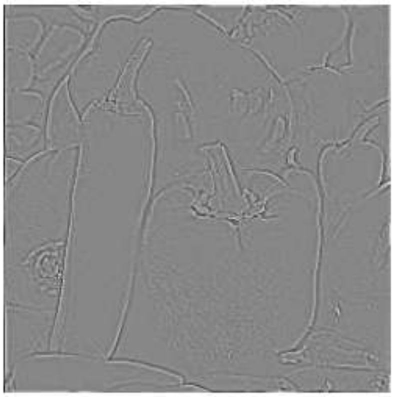



Butterworth HPF with $r_0 = 47$ and 2

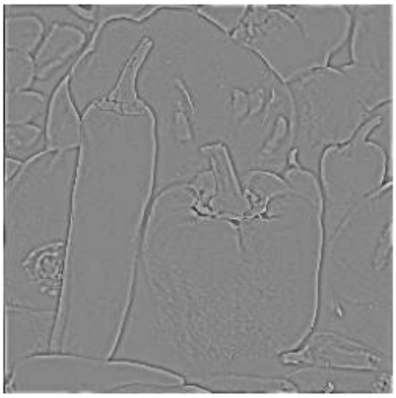
- Because the frequency response does not have a sharp transition like the ideal HPF, it is better for image sharpening because it does not introduce ringing.

**Butterworth HPF example**

Original Image



HPF image, $r_0 = 47$



High-pass Gaussian filters

• In two dimensions, the form of a Gaussian lowpass filter is 2 2 (,) / 2 (,) 1 − σ = -D u v H u v e, where 2 2 D(u,v) = u + v is the distance from the origin in the frequency plane.

The greater the value of, the higher the cutoff frequency and the harsher the filtering

## 3.4.3 HOMOMORPHIC FILTER

***HOMOMORPHIC FILTERING***

Light reflected from objects is used to create images. The image F(x,y) has two basic characteristics: (1) the amount of source light incident on the scene being viewed, and (2) the amount of light reflected by the objects in the scene. The illumination and reflectance components of light are indicated by the letters i(x,y) and r(x,y), respectively. The image function F is created by multiplying the functions i and r:

$$F(x,y) = i(x,y)r(x,y),$$

where $0 < i(x, y) < \infty$ and $0 < r(x,y) < 1$. We cannot easily use the above product to operate separately on the frequency components of illumination and reflection because the Fourier transform of the product of two functions is not separable; that is

$$\mathcal{F}(F(x, y)) \neq \mathcal{F}(i(x, y))\mathcal{F}(r(x, y)).$$

Let's say, on the other hand, that we define

$$\begin{aligned} z(x, y) &= \ln F(x, y) \\ &= \ln i(x, y) + \ln r(x, y). \end{aligned}$$

Then

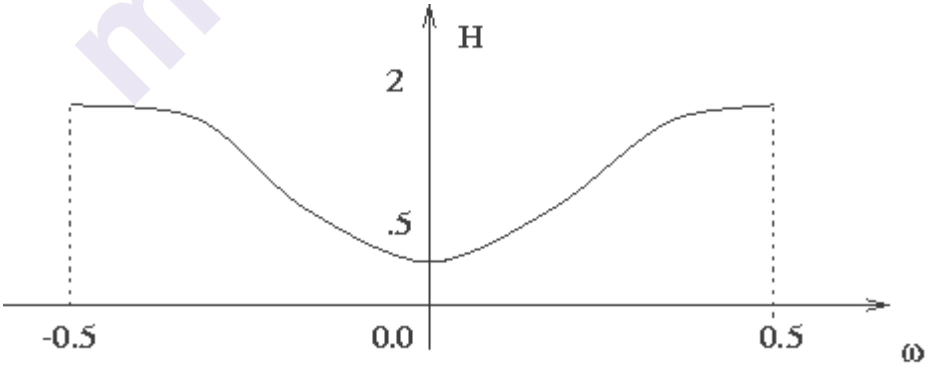
$$\begin{aligned} \mathcal{F}(z(x, y)) &= \mathcal{F}(\ln F(x, y)) \\ &= \mathcal{F}(\ln i(x, y)) + \mathcal{F}(\ln r(x, y)) \end{aligned}$$

or

$$Z(\omega, \nu) = I(\omega, \nu) + R(\omega, \nu),$$

The Fourier transforms of $z, \ln i$ and are Z, I, and R, respectively. The Fourier transform of the sum of two images: a low frequency illumination image and a high frequency reflectance image is represented by the function Z.

**Figure 6:** Transfer function for homomorphic filtering.




We may now suppress the light component while enhancing the reflectance component by using a filter with a transfer function that suppresses low frequency components while enhancing high frequency components. Thus

$$\begin{aligned} S(\omega, \nu) &= H(\omega, \nu)Z(\omega, \nu) \\ &= H(\omega, \nu)I(\omega, \nu) + H(\omega, \nu)R(\omega, \nu), \end{aligned}$$

Where S is the result's Fourier transform. In the realm of space,

$$\begin{aligned} s(x,y) &= \mathcal{F}^{-1}(S(\omega,\nu)) \\ &= \mathcal{F}^{-1}(H(\omega,\nu)I(\omega,\nu)) + \mathcal{F}^{-1}(H(\omega,\nu)R(\omega,\nu)). \end{aligned}$$

By letting

$$i'(x,y) = \mathcal{F}^{-1}(H(\omega,\nu)I(\omega,\nu))$$

and

$$r'(x,y) = \mathcal{F}^{-1}(H(\omega,\nu)R(\omega,\nu))$$

We get

$s(x,y) = i'(x,y) + r'(x,y)$.

Finally, because z was calculated by taking the logarithm of the original image F, the inverse produces the desired augmented image:

$$\begin{aligned} \hat{F}(x,y) &= exp[s(x,y)] \\ &= exp[i'(x,y)]exp[r'(x,y)] \\ &= i_0(x,y)r_0(x,y). \end{aligned}$$

As a result, the following figure can be used to summarise the homomorphic filtering process:

**Figure 7:** The process of homomorphic filtering.


## 3.5 LET US SUM UP

The Spatial Filtering technique is applied to individual pixels in an image. A mask is typically thought to be increased in size so that it has a distinct center pixel. Average (or mean) filtering is a technique for smoothing photographs by lowering intensity fluctuation between adjacent pixels. The best picture enhancement strategy is determined by the type of noise as well as the amount and level of noise in an image. Both linear and nonlinear actions are possible with our filters.

We will concentrate on two strategies based on the notion of filtering an original image. The averaging filter is used to reduce image detail. This filtering preserves the edges better. A sharpening spatial filter serves the exact opposite objective as the smoothing spatial filter. Its primary goal is to eliminate blurring and highlight the edges. The first and second-order derivatives are used.

## 3.6 LIST OF REFERENCES


1. **https://www.geeksforgeeks.org/spatial-filtering-and-its-types/**
2. http://www.seas.ucla.edu/dsplab/ie/over.html
3. **https://www.geeksforgeeks.org/spatial-filtering-and-its-types/**
4. https://www.theobjects.com/dragonfly/dfhelp/3-5/Content/05_Image%20Processing/Smoothing%20Filters.htm#:~:text=Mean%20filtering%20is%20a%20simple,of%20its%20neighbors%2C%20including%20itself.
5. http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-tomean-shift-algorithm/
6. **https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node4.html#:~:text=Image%20enhancement%20in%20the%20frequency,to%20produce%20the%20enhanced%20image.**
7. file:///E:/MY%20IMP%20documents/Lecture_9.pdf
8. file:///E:/MY%20IMP%20documents/Lecture_9.pdf


## 3.7 BIBLIOGRAPHY


1. **https://www.geeksforgeeks.org/spatial-filtering-and-its-types/**
2. http://www.seas.ucla.edu/dsplab/ie/over.html
3. **https://www.geeksforgeeks.org/spatial-filtering-and-its-types/**
4. https://www.theobjects.com/dragonfly/dfhelp/3-5/Content/05_Image%20Processing/Smoothing%20Filters.htm#:~:text=Mean%20filtering%20is%20a%20simple,of%20its%20neighbors%2C%20including%20itself.
5. http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-tomean-shift-algorithm/
6. **https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node4.html#:~:text=Image%20enhancement%20in%20the%20frequency,to%20produce%20the%20enhanced%20image.**
7. file:///E:/MY%20IMP%20documents/Lecture_9.pdf
8. file:///E:/MY%20IMP%20documents/Lecture_9.pdf
9. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node4.html#:~:text=Image%20enhancement%20in%20the%20frequency,to%20produce%20the%20enhanced%20image.

## 3.8 UNIT END EXERCISES

Q1. Why does the averaging filter cause the image to blur?

Q2. How does applying an average filter to a digital image affect it?

Q3. What does it mean to sharpen spatial filters?

Q4. What is the primary purpose of image sharpening?

Q5. What is the best way to sharpen an image?

Q6. How do you figure out what a low-pass filter's cutoff frequency is?

Q7. What is the purpose of a low-pass filter?

Q8. What is the effect of high pass filtering on an image?

Q9. In homomorphic filtering, which filter is used?

Q10. In homomorphic filtering, which high-pass filter is used?

❖❖❖❖

# Module III

# 4

# DISCRETE FOURIER TRANSFORM-I

**Unit Structure**



## 4.1 OBJECTIVES

After going through this unit, you will be able to:

- Understood the fundamental concepts of Digital Image processing
- Able to discuss mathematical transforms.
- Describe the DCT and DFT techniques
- Classify different types of image transforms
- Examine the use of Fourier transforms for image processing in the frequency domain

## 4.2 INTRODUCTION

In the realm of image processing, the Fourier transform is commonly employed. An image is a function that varies in space. Decomposing an image into a series of orthogonal functions, one of which being the Fourier functions, is one technique to analyse spatial fluctuations. An intensity image is transformed into the spatial frequency domain using the Fourier transform.

The sampling process converts a continuous-time signal x(t) into a discrete-time signal x(nT), where T is the sample interval.

x(t) sampling to x(nT)

The Fourier transform of a finite energy discrete time signal x(nT) is given by [1]

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\Omega nt}$$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j(\Omega t)n}$$

where $X(e^{j\omega})$ is a continuous function of ω and is known as Discrete-Time Fourier Transform (DTFT).

The relationship between ω and Ω is defined by

$\omega = \Omega T$

Replacing Ω by 2πf

$\omega = 2\pi f \times T$

where T is the sampling interval and is equal to 1/fs. Replacing T by 1/fs

$\omega = 2\pi f \times 1/fs$

where fs is the sampling frequency

$$\frac{f}{fs} = k$$

$\omega = k \times 2\pi$

To limit the infinite number of values to a finite number, Eq. is modified as

$$\frac{w}{2\Pi} = \frac{k}{N}$$

The Discrete Fourier Transform (DFT) of a finite duration sequence x(n) is defined as

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\Pi}{N}kn}$$

where k = 0, 1......, N – 1

The discrete-frequency representation (DFT) transfers a discrete signal onto a complex sinusoidal basis.

## 4.3 PROPERTIES OF DFT

We checked the periodicity of the combination by applying the DFT on a combination of two periodic sequences x1(n), x2(n). Because DFT is defined over a single period, the DFT of combination must have a single periodicity to be well described. In the continuous example, there are three types of combinations: linear ax1+bx2, convolution of x1 & x2, and multiplication x1 x2. For the continuous case, x1(n) is combined with x2 to define both linear combination and multiplication (n). Similarly, each x1(i) in the discrete case should be coupled with x2 (i). As a result, x1(n) and x2(n) have the same periodicity N, and the resultant series has the same periodicity N. If two sequences have distinct periodicities N1 and N2, padding transforms the periodicity N1 sequence into periodicity N2 by adding zeros at the end of N1.

i) **Linearity Property** :

Let X1(k) = DFT of x1(n) & X2(k) = DFT of x2(n)

∴ DFT {a x1(n) + b x2(n) } = a X1(k) + b X2(k) where a,b are

constants.

ii) **Periodicity** :

If a sequence x(n) periodic with periodicity N then N point DFT, X(k) is also periodic with periodicity N .

Let x(n+N ) = x(n) $\forall\, n$.

Then DFT (X (k+N)) = X(k) $\forall\, k$

iii) **Circular Time shift** :

It states that if discrete time signal is circularly shifted in time by m units then it's DFT is multiplied by $e^{\frac{-j2\Pi km}{N}}$

If DFT x(n) = X(k) Then DFT $\{(x(n-m) \bmod N\} = X(k)e^{\frac{-j2\Pi km}{N}}$

iv) **Circular Frequency shift** :

If discrete time signal multiplied by $e^{\frac{j2\Pi nm}{N}}$

then DFT is circularly shifted by m units.

If DFT x(n) = X(k) Then DFT $\{x(n)e^{\frac{j2\Pi nm}{N}}\} = X((k-m))N$

**v) Multiplication** :

DFT of product of two discrete time sequence equivalent to circular

convolution of DFT of individual sequences scaled by factor $1/N$ .

If DFT x(n) = X (k), Then DFT {x1(n) x2 (n)}= $1/N \; \{X1(k) \circledast X2(k)\}$

## 4.4 FFT ALGORITHMS Ñ DIRECT, DIVIDE AND CONQUER APPROACH [2]

DFT calculation is made more efficient using FFT algorithms. The method, which employs a divide-and-conquer strategy, reduces a DFT of size N, where N is a composite number, to the computation of smaller DFTs from which the bigger DFT is computed. We describe essential computational strategies, known as fast Fourier transform (FFT) algorithms, for computing the DFT when the size N is a power of two or power of four.

According to the formula, the computing challenge for the DFT is to compute the sequence {X(k)} of N complex-valued integers given another sequence of data x(n) of length N.

$$X(K) = \sum_{n=0}^{N-1} W_N^{kn} \qquad 0 \le k \le N-1$$

$$W_N = e^{-j2\pi/N}$$

where

Similarly, IDFT given as,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, \qquad 0 \le n \le N-1$$

We see that direct computation of X(k) requires N complex multiplications (4N real multiplications) and N —1 complex adds (4N — 2 real additions) for each value of k. As a result, computing all N DFT values necessitates N2 complex multiplications and N — N complex additions.

Direct DFT computation is inefficient primarily because it does not take advantage of the phase factor IN's symmetry and periodicity features. These two properties in particular are:

Property of symmetry: $W_N^{k+N/2} = -W_N^k W_N^{k+N/2} = -W_N^k$

Property of periodicity: $W_N^{k+N} = W_N^k W_N^{k+N} = W_N^k$

These two essential features of the phase factor are used by the computationally efficient algorithms presented in this section, commonly known as fast Fourier transform (FFT) algorithms.

### 4.4.1 Direct Computation of the DFT

For a complex-valued sequence x*(n)* of *N* points, the DFT may be expressed as

$$X_R(k) = \sum_{n=0}^{N-1} [x_R(n) \cos \frac{2\Pi kn}{N} + x_I(n) \sin \frac{2\Pi kn}{N}$$

$$X_I(k) = -\sum_{n=0}^{N-1} [x_R(n) \sin \frac{2\Pi kn}{N} - x_I(n) \cos \frac{2\Pi kn}{N}$$

The direct computation of above equations requires:

- $2N^2$ evaluations of trigonometric functions.
- $4N^2$ real multiplications.

- 4N(N-1) real additions.
- A number of indexing and addressing operations.

These are common operations in DFT computational techniques. The DFT values $X_R(k)$ and $X_I(k)$ are obtained by the procedures in items 2 and 3. To retrieve the data x(n), 0 to N - 1, and the phase factors, as well as to store the results, indexing and addressing procedures are required. Each of these computing operations is optimized differently by the various DFT methods.

**4.4.2 Divide-and-Conquer Approach to Computation of the DFT**

If we take a divide-and-conquer method, we can design computationally efficient DFT algorithms. This method is based on decomposing an N-point DFT into smaller and smaller DFTs. The FFT algorithms are a class of computationally efficient algorithms based on this basic principle.

To illustrate the computation of an *N* -point DFT, where *N* can be factored as a product of two integers, that is,

N =LM

Because we can pad any sequence with zeros to secure a factorization of the form above equation, the condition that N is not a prime integer is not limiting.

As shown in Fig. 1, the sequence x(n), 0< n< N —1, can be stored in a one-dimensional array indexed by nor a two-dimensional array indexed by 1 and m, where 0 <l <L — 1 and 0< m< M — 1 respectively.

The row index is /, but the column index is m.

As a result, the sequence x(n) can be saved in a rectangular format.

| $n \rightarrow$ | 0 | 1 | | $\cdots$ | $N-1$ |
|---|---|---|---|---|---|
| | $x(0)$ | $x(1)$ | $x(2)$ | $\cdots$ | $x(N-1)$ |

(a)

| row index $l$ \ $m$ column index | 0 | 1 | | $M-1$ |
|---|---|---|---|---|
| 0 | $x(0, 0)$ | $x(0, 1)$ | $\cdots$ | |
| 1 | $x(1, 0)$ | $x(1, 1)$ | $\cdots$ | |
| 2 | $x(2, 0)$ | $x(2, 1)$ | $\cdots$ | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ |
| $L-1$ | | | $\cdots$ | |

(b)

**Fig. 1 Two dimensional data array for storing the sequence x(n) 0 < n < *N-1***

array in a variety of ways, each of which depends on the mapping of index n to the " indexes (l, m).

For example, suppose that we select the mapping

n = Ml + m

This leads to an arrangement in which the first row consists of the first M elements of x(n), the second row consists of the next M elements of x(n), and so on, as illustrated in Fig. 2(a). On the other hand, the mapping

n = 1 + mL

stores the first L elements of x(n) in the first column, the next L elements in the second column, and so on, as illustrated in Fig.2(b).

Row-wise $n = Ml + m$

| $l$ \ $m$ | 0 | 1 | 2 | | $M-1$ |
|---|---|---|---|---|---|
| 0 | $x(0)$ | $x(1)$ | $x(2)$ | $\cdots$ | $x(M-1)$ |
| 1 | $x(M)$ | $x(M+1)$ | $x(M+2)$ | $\cdots$ | $x(2M-1)$ |
| 2 | $x(2M)$ | $x(2M+1)$ | $x(2M+2)$ | $\cdots$ | $x(3M-1)$ |
| | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ |
| $L-1$ | $x((L-1)M)$ | $x((L-1)M+1)$ | $x((L-1)M+2)$ | $\cdots$ | $x(LM-1)$ |

(a)

Column-wise $\quad n = l + mL$

| $l \backslash m$ | 0 | 1 | 2 | | $M-1$ |
|---|---|---|---|---|---|
| 0 | $x(0)$ | $x(L)$ | $x(2L)$ | $\cdots$ | $x((M-1)L)$ |
| 1 | $x(1)$ | $x(L+1)$ | $x(2L+1)$ | $\cdots$ | $x((M-1)L+1)$ |
| 2 | $x(2)$ | $x(L+2)$ | $x(2L+2)$ | $\cdots$ | $x((M-1)L+2)$ |
| | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ |
| $L-1$ | $x(L-1)$ | $x(2L-1)$ | $x(3L-1)$ | $\cdots$ | $x(LM-1)$ |

(b)

**Fig. 2 Two arrangements for the data arrays**

The computed DFT values can be stored in a similar manner.

The mapping is specifically from the index k to a pair of indices (p, q), with 0 <p < L - 1 and 0 <q < M - 1.

The DFT is stored on a row-by-row basis if the mapping

K = Mp+q

is chosen, with the first row containing the first M elements of the DFT X(k), the second row containing the following set of M elements, and so on.

The mapping

k = qL+ p,

leads in column-wise X(k) storage, with the first L elements stored in the first column, the second set of L elements in the second column, and so on.

Assume that x(n) is mapped to the rectangular array x(l, m) and that X(k) is mapped to a comparable rectangular array X(p, q).

The DFT can therefore be written as a double sum over the rectangle array's elements multiplied by the phase factors. Then,

$$X(p, q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m) W_N^{(Mp+q)(mL+l)}$$

,

But

$$W_N^{(Mp+q)(mL+l)} = W_N^{MLmp} W_N^{mLq} W_N^{Mpl} W_N^{lq}$$

$$W_N^{Nmp} = 1,\ W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq},\ \text{and}\ W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl}$$

$$X(p, q) = \sum_{l=0}^{L-1} \left\{ W_N^{lq} \left[ \sum_{m=0}^{M-1} x(l, m) W_M^{mq} \right] \right\} W_L^{lp}$$

The expression involves the computation of DFTs of length *M* and length *L*. To elaborate, let us subdivide the computation into three steps:

i) we compute the M-point DFTs

$$F(l, q) \equiv \sum_{m=0}^{M-1} x(l, m) W_M^{mq}, \qquad 0 \leq q \leq M - 1$$

for each of the rows l = 0, 1, ... , L — 1.

ii) we compute a new rectangular array G(l, q) defined as

$$G(l, q) = W_N^{lq} F(l, q), \qquad \begin{array}{l} 0 \leq l \leq L - 1 \\ 0 \leq q \leq M - 1 \end{array}$$

iii) Finally, we compute the L-point DFTs

$$X(p, q) = \sum_{l=0}^{L-1} G(l, q) W_L^{lp}$$

for each column q = 0, 1, ... ,M - 1, of the array G(1, q).

On the surface, the computing process given above appears to be more complicated than the direct DFT computation. The first phase entails computing L DFTs with M points each. As a result, LM complex multiplications and LM(M - 1) complex additions are required in this phase. The second phase necessitates the application of LM complex multiplications. Finally, MLV complex multiplications and ML(L - 1) complex additions are required in the third step of the algorithm. As a result, the computational difficulty is

Complex multiplications: N(M + L + 1)

Complex additions: N(M + L - 2)

where N = ML.

As a result, the number of multiplications has decreased from $N^2$ to N(M + L + 1), while the number of additions has decreased from N(N - 1) to N(M + L - 2).

To summarize, the algorithm that we have introduced involves the following computations:

**Algorithm 1**

1. Store the signal column-wise.
2. Compute the M-point DFT of each row.

3. Multiply the resulting array by the phase factors $W_N^{lq}$

4. Compute the $L$ -point DFT of each column

5. Read the resulting array row-wise.

An additional algorithm with a similar computational structure can be obtained if the input signal is stored row-wise and the resulting transformation is column-wise. This case we select,

n = Ml + m

k = qL + p

This choice of indices leads to the formula for the DFT in the form,

$$X(p,q) = \sum_{m=0}^{M-1}\sum_{l=0}^{L-1} x(l,m) W_N^{pm} W_L^{pl} W_M^{qm}$$

$$= \sum_{m=0}^{M-1} W_M^{mq} \left[\sum_{l=0}^{L-1} x(l,m) W_L^{lp}\right] W_N^{mp}$$

Thus we obtain a second algorithm.

**Algorithm 2**

1. Store the signal row-wise.

2. Compute the $L$ -point DFT at each column.

3. Multiply the resulting array by the factors $W_N^{m} W_N^{m}$

4. Compute the M-point DFT of each row.

5. Read the resulting array column-wise.

## 4.5 2D DISCRETE FOURIER TRANSFORM (DFT) AND FAST FOURIER TRANSFORM (FFT)[1]:

### 3.1.5.1 2D Discrete Fourier Transform (DFT) :

The 2D-DFT of a rectangular image f(m, n) of size M × N is represented as F(k, l)

f (m, n)----2D DFT→F(k, l)

where F(k, l) is defined as

$$F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\Pi}{M}mk} e^{-j\frac{2\Pi}{N}nl}$$

$$F(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\Pi}{M}nl} e^{-j\frac{2\Pi}{N}mk}$$

For a square image f (m, n) of size N × N, the 2D DFT is defined as

$$F(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\, e^{-j\frac{2\Pi}{N}mk} e^{-j\frac{2\Pi}{N}nl}$$

The inverse 2D Discrete Fourier Transform is given by

$$f(m,n) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F(k,l)\, e^{j\frac{2\Pi}{N}mk} e^{j\frac{2\Pi}{N}nl}$$

The Fourier transform F (k, l) is given by

F(k,l) = R(k,l) + jI(k,l)

where R(k, l ) represents the real part of the spectrum and I(k, l) represents the imaginary part.

The Fourier transform F (k, l ) can be expressed in polar coordinates as

F( k,l)= mod ( F(k,l)) $e^{j\square kl}$

where mod (F(k, l)) = $(R^2\{F(k, l)\}+ I^2\{F(k, l)\})^{1/2}$ is called the magnitude spectrum of the Fourier transform and

$$\Psi(k,l) = \tan^{-1}\frac{I\{F(k,l)\}}{R\{F(k,l)\}}$$

is the phase angle or phase spectrum. Here, R{F(k, l)}, I{F(k, l)} are the real and imaginary parts of F(k, l) respectively.

The Fast Fourier Transform is the most computationally efficient type of DFT (FFT).

The FFT of an image can be represented in one of two ways: (a) conventional representation or (b) optical representation.

High frequencies are collected at the centre of the image in the standard form, whereas low frequencies are distributed at the edges, as seen in Fig. 1. The null frequency can be seen in the upper-left corner of the graph.

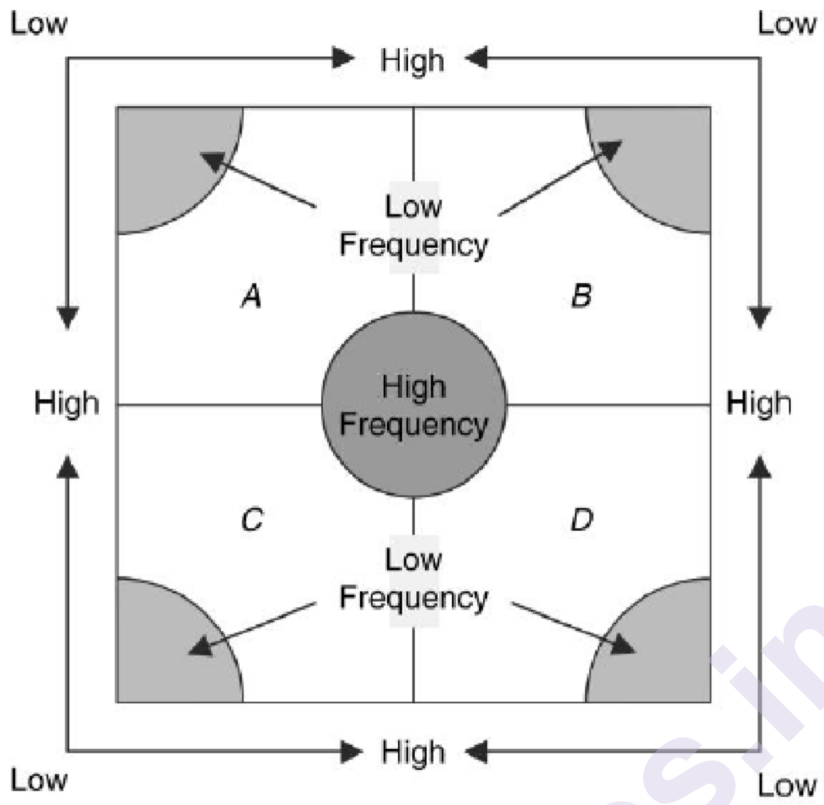



**Fig. 1 – Standard representation of FFT of an image [1,3]**

The frequency range is [0, N] X [0, M], where M is the image's horizontal resolution and N is the image's vertical resolution.

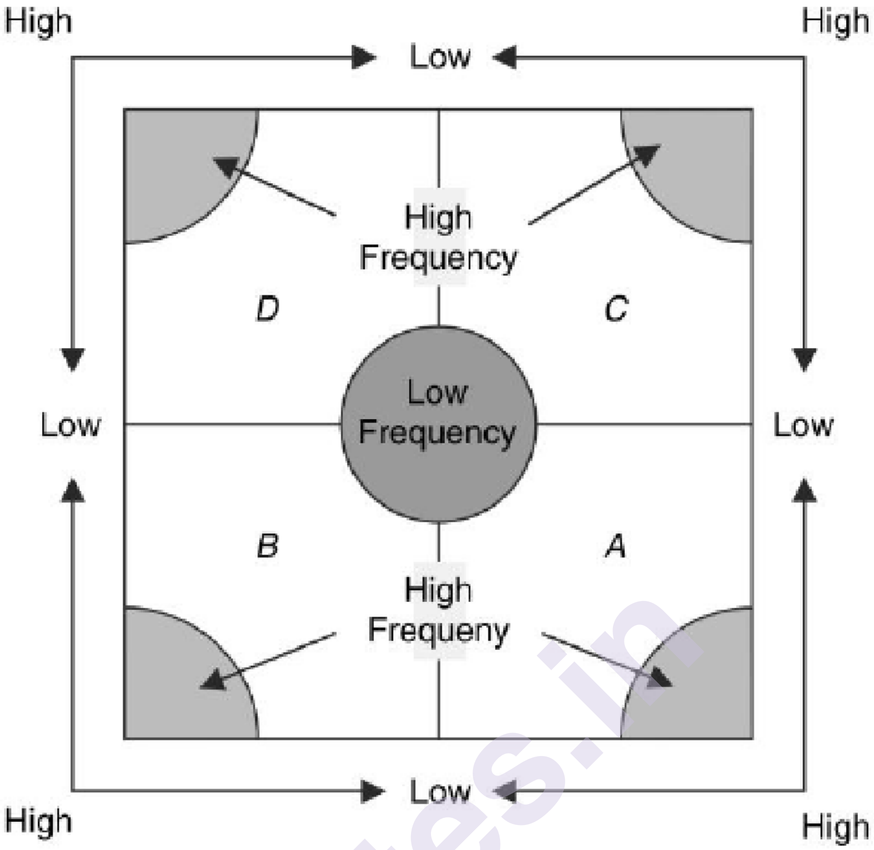



**Fig. 2 optical representation of the FFT of the same image.**

Discreteness in one domain leads to periodicity in another as in Fig. 2, as we all know. As a result, the spectrum of a digital image will be unique in the range – π to π or between 0 and 2π.

## 4.5.2 Computational speed of FFT [4]:

The DFT requires $N^2$ complex multiplications. At each stage of the FFT (i.e. each halving) N/2 complex multiplications are required to combine the results of the previous stage. Since there are ($\log_2 N$) stages, the number of complex multiplications required to evaluate an -point DFT with the FFT is approximately N/2 $\log_2 N$.

| $N$ | $N^2$ (DFT) | $\frac{N}{2}log_2N$ (FFT) | saving |
|---|---|---|---|
| 32 | 1,024 | 80 | 92% |
| 256 | 65,536 | 1,024 | 98% |
| 1,024 | 1,048,576 | 5,120 | 99.5% |

## 4.5.3 Practical considerations [4] :

If N is not a power of 2, there are 2 strategies available to complete N - point FFT.

1. take advantage of such factors as N possesses. For example, if N is divisible by 3‰ (e.g. N=48), the final decimation stage would include a ‰3 -point transform.

2. pack the data with zeroes; e.g. include 16 zeroes with the 48 data points (for N=48) and compute a 64-point FFT. (However, you should again be wary of abrupt transitions between the trailing (or leading) edge of the data and the following (or preceding) zeroes; a better approach might be to pack the data with more realistic "dummy values"). Zero padding cannot improve the resolution of spectral components, because the resolution is "proportional" to 1/M rather than 1/N. Zero padding is very important for fast DFT implementation (FFT).

## 4.6 SUMMARY :

Frequency smoothing and frequency leaking are examples of DFT applications on finite pictures with MxN pixels. DFT is based on discretely sampled pictures (pixels), which suffer from aliasing. DFT takes into account periodic boundary conditions including centering, edge effects, and convolution. Images have borders and are truncated (finite), resulting in frequency smoothing and leakage. All drawbacks of DFT overcomes by FFT.

## 4.7 REFERENCES

1] S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN- 13:978-0-07- 0144798

2] John G. Proakis, Digital Signal Processing: Principles, Algorithms, And Applications, 4/E

3] Gonzalez, Woods & Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191

4] https://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf

## 4.8 UNIT END EXERCISES

1. Find the N × N point DFT of the following 2D image f(m, n), $0 \le m, n \le N$

2. Prove that DFT diagonlises the circulant matrix.

3. Which of the following is true regarding the number of computations required to compute an N-point DFT?

a) N2 complex multiplications and N(N-1) complex additions
b) N2 complex additions and N(N-1) complex multiplications
c) N2 complex multiplications and N(N+1) complex additions
d) N2 complex additions and N(N+1) complex multiplications

**Answer : a**

4. Which of the following is true regarding the number of computations required to compute DFT at any one value of 'k'?

a) 4N-2 real multiplications and 4N real additions
b) 4N real multiplications and 4N-4 real additions
c) 4N-2 real multiplications and 4N+2 real additions
d) 4N real multiplications and 4N-2 real additions

**Answer : d**

5. Divide-and-conquer approach is based on the decomposition of an N-point DFT into successively smaller DFTs. This basic approach leads to FFT algorithms.

a) True
b) False

**Answer : a**

6. How many complex multiplications are performed in computing the N-point DFT of a sequence using divide-and-conquer method if N=LM?

a) N(L+M+2)
b) N(L+M-2)
c) N(L+M-1)
d) N(L+M+1)

**Answer : d**

7. Define discrete Fourier transform and its inverse.

8. State and prove the translation property.

9. Give the drawbacks of DFT.

10. Give the property of symmetry and Periodicity of Direct DFT.

❖❖❖❖

# 5

# DISCRETE FOURIER TRANSFORM-II

**Unit Structure**



## 5.1 OBJECTIVES

After going through this unit, you will be able to:

- Understood the fundamental concepts of Digital Image processing
- Able to discuss mathematical transforms.
- Describe the DCT and DFT techniques
- Classify different types of image transforms
- Examine the use of Fourier transforms for image processing in the frequency domain

## 5.2 INTRODUCTION



### 5.2.1 Image Transforms

A representation of an image is called as Image transform. The reasons for transforming an image from one representation to another are as-

i. The transformation may isolate critical components of the image pattern so that they are directly accessible for analysis.

ii. The transformation may place the image data in a more compact form so that they can be stored and transmitted efficiently.

### 5.2.2 Unitary Transform [1] :

A discrete linear transform is unitary if its transform matrix conforms to the unitary condition

$A \times A^H = I$

where A = transformation matrix, $A^H$ represents Hermitian matrix.

$A^H = A^{*T}$

I = identity matrix

When the transform matrix A is unitary, the defined transform is called unitary transform.

Example) Check whether the DFT matrix is unitary or not [1].

Step 1 : Determination of the matrix A

Finding 4-point DFT (where N = 4)

The formula to compute a DFT matrix of order 4 is given below

$$X(K) = \sum_{n=0}^{3} x(n)e^{-j\frac{2\Pi}{4}kn}$$

where k = 0, 1..., 3

1. Finding X(0)

$$X(0) = \sum_{n=0}^{3} x(n) = x(0) + x(1) + x(2) + x(3)$$

2. Finding X(1)

$$X(1) = \sum_{n=0}^{3} x(n)e^{-j\frac{\Pi}{2}n}$$

$$= x(0) + x(1)e^{-j\frac{\Pi}{2}} + x(2)e^{-j\Pi} + x(3)e^{-j\frac{3\Pi}{2}}$$

$X(1)$ = x(0)− jx(1)−x(2)+ jx(3)

3. Finding X(2)

$$X(2) = \sum_{n=0}^{3} x(n)e^{-j\Pi n}$$

$$= x(0) + x(1)e^{-j\Pi} + x(2)e^{-j2\Pi} + x(3)e^{-j3\Pi}$$

X(2) = x(0)−x(1)+ x(2)−x(3)

4. Finding X(3)

$$X(3) = \sum_{n=0}^{3} x(n)e^{-j\frac{3\Pi}{2}n}$$

$$= x(0) + x(1)e^{-j\frac{3\Pi}{2}} + x(2)e^{-j3\Pi} + x(3)e^{-j\frac{9\Pi}{2}}$$

X (3) = x(0)+ jx(1)−x(2)− jx(3)

Collecting the coefficients of X(0), X(1), X(2) and X(3), we get

$$X[K] = A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{vmatrix}$$

**Step 2 :** Computation of $A^H$

To determine $A^H$, first determine the conjugate and then take its transpose.

A Conjugate $A^*$ Transpose $A^H$

Step 2a : Computation of conjugate $A^*$

$$A^* = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{vmatrix}$$

Step 2b : Determination of transpose of $A^*$

$$( A^{*})^{T} = A^{H} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{vmatrix}$$

Step 3 Determination of $A \times A^{H}$

$$A \times A^{H} = x = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{vmatrix} \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{vmatrix}$$

$$= \begin{vmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{vmatrix}$$

$$= 4 \quad x \quad \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

.

The result is the identity matrix, which shows that Fourier transform satisfies unitary condition.

**Sequency** - It refers to the number of sign changes. The sequency for a DFT matrix of order 4 is given below.

**Transform Coefficients**

$$X[K] = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{vmatrix} \quad \begin{matrix} \text{Zero sign change} \\ \text{One sign changes} \\ \text{Two sign changes} \\ \text{Three sign changes} \end{matrix}$$

## 5.3 PROPERTIES OF 2-D DFT [1] :

The properties of 2D DFT are shown in table 1.

| Property | Sequence | Transform |
|---|---|---|
| Spatial shift | $f(m-m_0, n)$ | $e^{-j\frac{2\pi}{N}m_0k}F(k,l)$ |
| Periodicity | – | $F(k+pN, l+qN)=F(k,l)$ |
| Convolution | $f(m,n)*g(m,n)$ | $F(k,l)\times G(k,l)$ |
| Scaling | $f(am,bn)$ | $\frac{1}{\lvert ab\rvert}F(k/a,l/b)$ |
| Conjugate symmetry | | $F(k,l)=F^*(-k,-l)$ |
| Multiplication by exponential | $e^{j\frac{2\pi}{N}mk_0}e^{j\frac{2\pi}{N}ml_0}f(m,n)$ | $F(k-k_0, l-l_0)$ |
| Rotation property | $f(r\cos(\theta+\theta_0), r\sin(\theta+\theta_0))$ | $F[R\cos(\Phi+\Phi_0), R\sin(\Phi+\Phi_0)]$ |
| Folding property | $f(-m,-n)$ | $F(-k,-l)$ |

**Table 1 - properties of 2D DFT [1]**

## 5.4 CLASSIFICATION OF IMAGE TRANSFORMS :

A) Walsh transform : transforms with non-sinusoidal orthogonal basis functions

B) Hadamard transform : transforms with non-sinusoidal orthogonal basis functions

C) Discrete cosine transform : transforms with orthogonal basis functions

D) Discrete wavelet transform

- Haar Transforms : transforms with non-sinusoidal orthogonal basis functions
- KL transform : transforms whose basis functions depend on the statistics of the input data

### 5.4.1 Walsh Transform [1] :

The representation of a signal by a set of orthogonal sinusoidal waveforms is known as Fourier analysis. The frequency components are the coefficients of this representation, and the waveforms are arranged by frequency. To express these functions, Walsh created a comprehensive set of orthonormal square-wave functions. The Walsh function's computational simplicity stems from the fact that it is a real function with only two possible values: +1 or –1.

The one-dimensional Walsh transform basis can be given by the following equation [1]:

$$g(n,k)=\frac{1}{N}\prod_{i=0}^{m-1}(-1)^{b_i(n)b_{m-1-i}(k)}$$

where n = time index,

k = frequency index

N = order

m = number bits to represent a number

bi(n) = i th (from LSB) bit of the binary value

n decimal number represented in binary.

The value of m is given by m = $\log_2$ N.

The two-dimensional Walsh transform of a function f (m, n) is given by[1],

$$F(k,l)=\frac{1}{N}\sum_m\sum_n f(m,n)\prod_{i=0}^{p-1}(-1)^{\left[b_i(m)b_{p-1-i}(k)+b_i(n)b_{p-1-i}(k)\right]}$$

Example) Find the 1D Walsh basis for the fourth-order system (N = 4).

the value of N is given as four. From the value of N, the value of m is calculated as N = 4;

m = $\log_2$ N

=$\log_2 4 = \log_2 2^2$

=$2*\log_2 2$

m = 2

In this, N = 4. So n and k have the values of 0, 1, 2 and 3. I varies from 0 to m–1. From the above computation, m = 2. So i has the value of 0 and 1. The construction of Walsh basis for N = 4 is given in Table 1.

When k or n is equal to zero, the basis value will be 1/N.

$$g(n,k)=\frac{1}{N};\ for \quad n=0, \quad \text{or} \quad k=0$$

| *Decimal value* | *Binary values* | |
|---|---|---|
| $n$ | $b_1(n)$ | $b_0(n)$ |
| 0 | $b_1(0)=0$ | $b_0(0)=0$ |
| 1 | $b_1(1)=0$ | $b_0(1)=1$ |
| 2 | $b_1(2)=1$ | $b_0(2)=0$ |
| 3 | $b_1(3)=1$ | $b_0(3)=1$ |

Table 1 : Construction of walsh basis for N = 4 [1]

**Sequency :** The Walsh functions may be ordered by the number of zero crossings or sequency, and the coefficients of the representation may be called sequency components. The sequency of the Walsh basis function for N = 4 is shown in Table 2.

| $k \backslash n$ | 0 | 1 | 2 | 3 | Sequency |
|---|---|---|---|---|---|
| 0 | 1/4 | 1/4 | 1/4 | 1/4 | Zero sign change (DC value) |
| 1 | 1/4 | 1/4 | −1/4 | −1/4 | One sign change |
| 2 | 1/4 | −1/4 | 1/4 | −1/4 | Three sign changes |
| 3 | 1/4 | −1/4 | −1/4 | 1/4 | Two sign changes |

*Calculating the value for g(1,2)*

Table 2 : Walsh transform basis for N = 4 [1]

$$g(2,1)=\frac{1}{4}\prod_{i=0}^{1}(-1)^{b_i(2)b_{m-1-i}(1)}$$

$$g(2,1)=\frac{1}{4}\left\{\left((-1)^{b_0(2)b_1(1)}\right)\times\left((-1)^{b_1(2)b_0(1)}\right)\right\}$$

$$g(2,1)=\frac{1}{4}\left\{\left((-1)^{0\times 0}\right)\times\left((-1)^{1\times 1}\right)\right\}$$

$$g(2,1)=\frac{1}{4}\{(1)\times(-1)\}$$

$$g(2,1)=-\frac{1}{4}$$

Likewise, all the values of the Walsh transform can be calculated. After the calculation of all values, the basis for N = 4 is given below [1].

$$g(n,k)=\begin{pmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\ +\frac{1}{4} & +\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ +\frac{1}{4} & -\frac{1}{4} & +\frac{1}{4} & -\frac{1}{4} \\ +\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & +\frac{1}{4} \end{pmatrix}$$

Note: When looking at the Walsh basis, every entity has the same magnitude (1/N), with the only difference being the sign (whether it is positive or negative). As a result, the following is a shortcut approach for locating the sign:

Step 1 Write the binary representation of n.

Step 2 Write the binary representation of k in the reverse order.

Step 3 Check for the number of overlaps of 1 between n and k.

Step 4 If the number of overlaps of 1 is

i) zero then the sign is positive

ii) even then the sign is positive

iii) odd then the sign is negative

**5.4.2 Hadamard Transform :**

The Hadamard transform is similar to the Walsh transform with the exception that the rows of the transform matrix are re-ordered.

The elements of a Hadamard transform's mutually orthogonal basis vectors are either +1 or –1, resulting in a minimal computing complexity in calculating the transform coefficients.

The following approach can be used to create Hadamard matrices for $N = 2^n$:

The order N = 2 Hadamard matrix is given as,

$$H_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

The Hadamard matrix of order 2N can be generated by Kronecker product operation:

$$\begin{vmatrix} H_N & H_N \\ H_N & -H_N \end{vmatrix} \quad H_{2N} =$$

Substituting N = 2 in above equation,

$$\begin{vmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{vmatrix} \quad H_4 =$$

| | | | |
|---|---|---|---|
| **1** | **1** | **1** | **1** |
| **1** | **-1** | **1** | **-1** |
| **1** | **1** | **-1** | **-1** |
| **1** | **-1** | **-1** | **1** |

=

Similarly, substituting N = 4 in $H_{2N}$ equation,

The Hadamard matrix of order $N = 2^n$ may be generated from the order two core matrix. It is not desirable to store the entire matrix.

**5.4.3 Discrete cosine transform :**

Members of a family of real-valued discrete sinusoidal unitary transforms are discrete cosine transforms. A discrete cosine transform is made up of a set of sampled cosine functions and a set of basis vectors. DCT is a signal compression technique that breaks down a signal into its fundamental frequency components.

If x[n] is the signal of length N, the Fourier transform of the signal x[n] is given by X[k] where,

$$X(K) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\Pi kn}{N}}$$

where k varies between 0 to N − 1.

Consider extending the signal x[n], which is indicated by xe[n], so that the expanded sequence has a length of 2N. There are two ways to expand the sequence x[n].

Consider the following sequence (original sequence) of length four: x[n] = [1, 2, 3, 4]. Fig. 1 depicts the original sequence. There are two ways to lengthen the sequence. By simply copying the original sequence again, as shown in Fig. 2, the original sequence can be extended.

As demonstrated in Fig. 2, the expanded sequence can be created by simply replicating the original sequence. The biggest disadvantage of this method is the variance in sample value between n = 3 and n = 4.

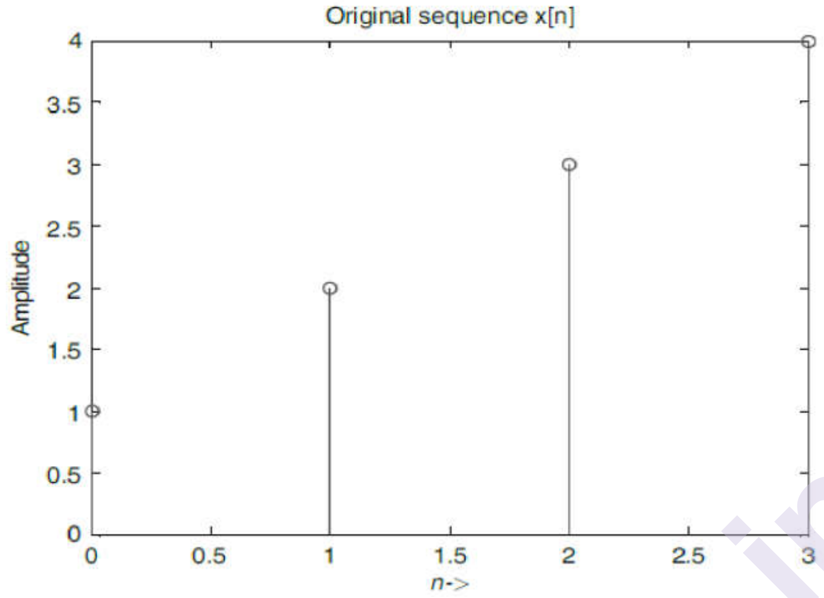



Fig. 1 Original sequence

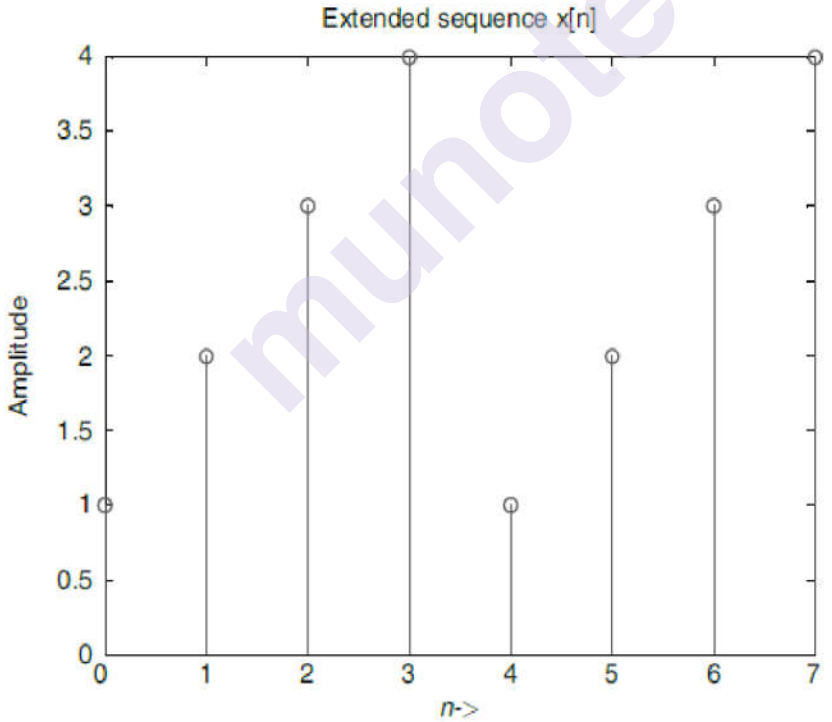



Fig. 2 Extended sequence obtained by simply copying the original sequence

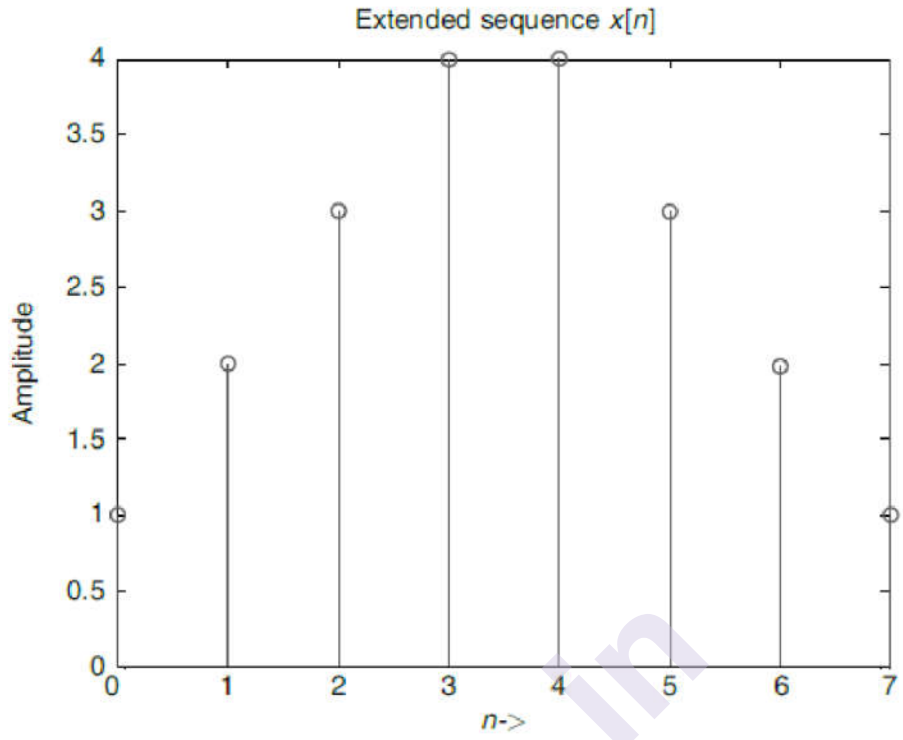



Fig. 3 Extended sequence obtained by folding the original sequence

The phenomena of 'ringing' is unavoidable due to the extreme fluctuation. A second approach of producing the expanded sequence, as illustrated in Fig. 3, is to copy the original sequence in a folded fashion. When comparing Figs. 2 and 3, it is obvious that the variance in the sample value at n = 3 and n = 4 in Fig. 3 is the smallest when compared to Fig. 2. The expanded sequence created by folding the initial sequence is shown to be a better choice as a result of this.

The length of the expanded sequence is 2N if N is the length of the original sequence, as seen in both Figs. 2 and 3.

In this example, the length of the original sequence is 4 (refer Fig. 1) and the length of the extended sequence is 8(refer Fig. 2 and Fig. 3).

The Discrete Fourier Transform (DFT) of the extended sequence is given by Xe[k] where

$$X_e(K) = \sum_{n=0}^{2N-1} x_e(n) e^{-j\frac{2\Pi kn}{2N}}$$

Split the interval 0 to 2N – 1 into two parts,

$$X_e(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\Pi kn}{2N}} + \sum_{n=N}^{2N-1} x[2N-1-n] e^{-j\frac{2\Pi kn}{2N}}$$

Let m = 2N – 1 − n. Substituting in above equation,

$$X_e(K) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\Pi kn}{2N}} + \sum_{m=0}^{N-1} x[m]e^{-j\frac{2\Pi k(2N-1-m)}{2N}}$$

$$X_e(K) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\Pi kn}{2N}} + \sum_{m=0}^{N-1} x[m]e^{-\frac{j2\Pi k}{2N}2N} e^{\frac{j2\pi}{2N}(m+1)}$$

But,

$$e^{-\frac{j2\pi k}{2N}2N} = e^{-j2\pi k} = \cos(2\pi k) - j\sin(2\pi k) = 1$$

$$X_e[k] = \sum_{n=0}^{N-1} x[n]e^{-\frac{j2\pi kn}{2N}} + \sum_{m=0}^{N-1} x[m]\, e^{\frac{j2\pi}{2N}(m+1)}$$

Replacing m by n and Multiplying both sides by $e^{-\frac{j\Pi k}{2N}}$

$$X_e[k]e^{\frac{-j\pi k}{2N}} = \sum_{n=0}^{N-1} x[n]\{e^{-\frac{j2\pi kn}{2N}} + e^{\frac{j2\pi(n+1)}{2N}}\}e^{\frac{-j\pi k}{2N}}$$

Upon simplification,

$$\frac{X_e[k]e^{\frac{-j\pi k}{2N}}}{2} = \sum_{n=0}^{N-1} x[n]\cos\left\{\frac{(2n+1)\pi k}{2N}\right\}$$

Thus, the kernel of a one-dimensional discrete cosine transform is given by

$$X[k] = \alpha(k)\sum_{n=0}^{N-1} x[n]\cos\left\{\frac{(2n+1)\pi k}{2N}\right\}, \text{ where } 0 \le k \le N-1$$

$$\alpha(k) = \sqrt{\frac{1}{N}} \text{ if } k = 0$$

$$\alpha(k) = \sqrt{\frac{2}{N}} \text{ if } k \ne 0$$

The process of reconstructing a set of spatial domain samples from the DCT coefficients is called the inverse discrete cosine transform (IDCT). The inverse discrete cosine transformation is given by,

$$x[n] = \alpha(k)\sum_{k=0}^{N-1} X[k]\cos\left[\frac{(2n+1)\pi k}{2N}\right], 0 \le n \le N-1$$

The forward 2D discrete cosine transform of a signal f(m, n) is given by,

$$F[k,l] = \alpha(k)\alpha(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)\cos\left[\frac{(2m+1)\pi k}{2N}\right]\cos\left[\frac{(2n+1)\pi l}{2N}\right]$$

$$\text{where } \alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } k=0 \\ \sqrt{\frac{2}{N}} & \text{if } k \ne 0 \end{cases}$$

$$\text{Similarly } \alpha(l) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } l=0 \\ \sqrt{\frac{2}{N}} & \text{if } l \ne 0 \end{cases}$$

The 2D inverse discrete cosine transform is given by

$$f[m,n] = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \alpha(k)\alpha(l)F(k,l)\cos\left[\frac{(2m+1)\pi k}{2N}\right]\cos\left[\frac{(2n+1)\pi l}{2N}\right]$$

**5.4.4 Discrete Wavelet Transform:** Haar Transform, KL Transform

**5.4.4.1 Haar Transform :**

The Haar transform is based on a class of orthogonal matrices with elements of 1, –1, or 0 multiplied by powers of √2 as its elements. The Haar transform is computationally efficient since it only requires 2(N – 1) additions and N multiplications to change an N-point vector.

**Algorithm to Generate Haar Basis** [1]: The algorithm to generate Haar basis is given below:

**Step 1** Determine the order of N of the Haar basis.

**Step 2** Determine n where n = $\log_2$ N.

**Step 3** Determine p and q.

(i) $0 \le p < n-1$

(ii) If p = 0 then q = 0 or q = 1

(iii) If $p \ne 0$, $1 \le q \le 2^p$

**Step 4** Determine k.

$k = 2^p + q - 1$

**Step 5** Determine Z.

$$Z \rightarrow [0, 1) \Rightarrow \left\{ \frac{0}{N}, \quad \frac{1}{N}, \quad \cdots \quad \frac{N-1}{N} \right\}$$

**Step 6** If k = 0 then $H(Z) = 1/\sqrt{N}$

Otherwise

,

$$H_k(Z) = H_{pq}(Z) = \frac{1}{\sqrt{N}} \begin{cases} +2^{p/2} & \dfrac{(q-1)}{2^p} \le Z < \dfrac{(q-1/2)}{2^p} \\ -2^{p/2} & \dfrac{(q-1/2)}{2^p} \le Z < \dfrac{q}{2^p} \\ 0 & \text{otherwise} \end{cases}$$

The flow chart to compute Haar basis is given Fig. 4

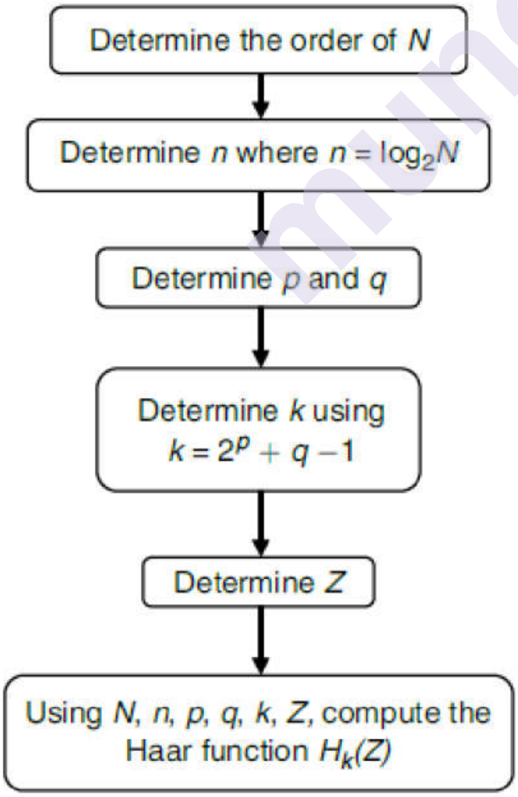



**Fig. 4 Flow chart to compute Haar basis**

### 5.4.4.2 KL Transform (KARHUNEN–LOEVE TRANSFORM) :

Harold Hotelling was the first to study the discrete formulation of the KL transform, which is why it is also known as the Hotelling transform. The KL transform is a reversible linear transform that takes advantage of a vector representation's statistical features.

The orthogonal eigenvectors of a data set's covariance matrix are the basic functions of the KL transform. The input data is optimally decorrelated using a KL transform. The majority of the 'energy' of the transform coefficients is focused inside the first few components after a KL transform. A KL transform's energy compaction property is this.

**Drawbacks of KL transform** :

i. A KL transform is input-dependent, and the fundamental function for each signal model on which it acts must be determined. There is no unique mathematical structure in the KL bases that allows for quick implementation.

ii. The KL transform necessitates multiply/add operations in the order of $O(m^2)$. $O(\log_2{}^m)$ multiplications are required for the DFT and DCT.

**Applications of KL Transforms [1] :**

(i) **Clustering Analysis** : Used to determine a new coordinate system for sample data where the largest variance of a projection of the data lies on the first axis, the next largest variance on the second axis, and so on.

(ii) **Image Compression** : It is heavily utilised for performance evaluation of compression algorithms since it has been proven to be the optimal transform for the compression of an image sequence in the sense that the KL spectrum contains the largest number of zero-valued coefficients.

Example) Perform KL transform for the following matrix: $X = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}$

Step 1- Formation of vectors from the given matrix

The given matrix is a 2×2 matrix; hence two vectors can be extracted from the given matrix. Let it be

x0 and x1.

$$x_0 = \begin{bmatrix} 4 \\ -1 \end{bmatrix} \text{ and } x_1 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

Step 2 Determination of covariance matrix

The formula to compute covariance of the matrix is $\operatorname{cov}(x) = E[xx^T] - \bar{x}\,\bar{x}^T$

In the formula for covariance matrix, x denotes the mean of the input matrix. The formula to compute

the mean of the given matrix is given below:

$$\bar{x} = \frac{1}{M}\sum_{k=0}^{M-1} x_k$$

where M is the number of vectors in x.

$$\bar{x} = \frac{1}{2}\sum_{k=0}^{1} x_k = \frac{1}{2}\{x_0 + x_1\} = \frac{1}{2}\left\{\begin{bmatrix}4\\-1\end{bmatrix} + \begin{bmatrix}-2\\3\end{bmatrix}\right\}$$

$$= \frac{1}{2}\left\{\begin{bmatrix}2\\2\end{bmatrix}\right\} = \begin{bmatrix}1\\1\end{bmatrix}$$

The mean value is calculated as

$$\bar{x} = \begin{bmatrix}1\\1\end{bmatrix}$$

Now multiplying the mean value with its transpose yields

$$\bar{x}\,\bar{x}^T = \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\end{bmatrix} = \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$$

To find the $E\left|xx^T\right|$

$$E\left[xx^T\right] = \frac{1}{M}\sum_{k=0}^{M-1} x_k x_k^T$$

In our case, M = 2 hence

$$E\left[xx^T\right] = \frac{1}{2}\sum_{k=0}^{1} x_k x_k^T = \frac{1}{2}\left\{\begin{bmatrix}4\\-1\end{bmatrix}\begin{bmatrix}4 & -1\end{bmatrix} + \begin{bmatrix}-2\\3\end{bmatrix}\begin{bmatrix}-2 & 3\end{bmatrix}\right\}$$

$$E\left[xx^T\right] = \frac{1}{2}\left\{\begin{bmatrix}16 & -4\\-4 & 1\end{bmatrix} + \begin{bmatrix}4 & -6\\-6 & 9\end{bmatrix}\right\} = \begin{bmatrix}10 & -5\\-5 & 5\end{bmatrix} = \begin{bmatrix}2 & -1\\-1 & 1\end{bmatrix}$$

Now using the value of $E\left[xx^T\right]$ and $\bar{x}\,\bar{x}^T$, we find the covariance matrix,

$$\text{cov}(x) = E[xx^T] - \bar{x}\,\bar{x}^T$$

$$\text{cov}(x) = \begin{bmatrix}2 & -1\\-1 & 1\end{bmatrix} - \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} = \begin{bmatrix}1 & -2\\-2 & 0\end{bmatrix}$$

Step 3 Determination of eigen values of the covariance matrix

To find the eigen values λ, we solve the characteristic equation,

$$\left|\mathrm{cov}(x) - \lambda I\right| = 0$$

$$\det\left(\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$\det\left(\begin{bmatrix} 1-\lambda & -2 \\ -2 & -\lambda \end{bmatrix}\right) = 0$$

$\lambda^2 - \lambda - 4 = 0$

From the last equation, we have to find the eigen values $\lambda_0$, $\lambda_1$. Solving above equation,

$$\lambda_0 = \frac{1+4.1231}{2} = 2.5615$$

$$\lambda_1 = \frac{1-4.1231}{2} = -1.5615$$

Step 4- Determination of eigen vectors of the covariance matrix

The first eigen vector $\varphi_0$ is found from the equation,

$$\left(\mathrm{cov}(x) - \lambda_0 I\right)\phi_0 = 0.$$

$$\left(\mathrm{cov}(x) - \lambda_0 I\right)\phi_0 = \left[\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} 2.5615 & 0 \\ 0 & 2.5615 \end{bmatrix}\right]\begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1.5615 & -2 \\ -2 & -2.5615 \end{bmatrix}\begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Using row detection method, we have found, $\phi_{01}$ to be a free variable. So, we choose the value of $\phi_{01}$ as 1.

$$-1.5615\phi_{00} - 2\phi_{01} = 0$$

$$\phi_{00} = \frac{2}{-1.5615} = -1.2808$$

The eigen vector $\phi_0 = \begin{bmatrix} -1.2808 \\ 1 \end{bmatrix}$

Similarly, find the next eigen vector $\phi_1$; the eigen value is $\lambda_1 = -1.5615$

$$(\mathrm{cov}(x) - \lambda_1 I)\phi_1 = \left[\begin{bmatrix} 1 & -2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} -1.5615 & 0 \\ 0 & -1.5615 \end{bmatrix}\right]\begin{bmatrix} \phi_{10} \\ \phi_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 2.5615 & -2 \\ -2 & -1.5615 \end{bmatrix}\begin{bmatrix} \phi_{10} \\ \phi_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Using row detection method, we have found $\phi_{11}$ to be a free variable. So choose, we the value of $\phi_{11}$ as 1.

$$2.5615\phi_{10} - 2\phi_{11} = 0$$

$$\phi_{10} = \frac{2}{2.5615} = 0.7808$$

The eigen vector $\phi_1 = \begin{bmatrix} 0.7808 \\ 1 \end{bmatrix}$

**Step 5** - Normalisation of the eigen vectors

The normalisation formula to normalise the eigen vector $\varphi_0$ is,

$$\frac{\phi_0}{\|\phi_0\|} = \frac{1}{\sqrt{\phi_{00}^2 + \phi_{01}^2}}\begin{bmatrix} \phi_{00} \\ \phi_{01} \end{bmatrix}$$

$$\frac{\phi_0}{\|\phi_0\|} = \frac{1}{\sqrt{(-1.2808)^2 + 1^2}}\begin{bmatrix} -1.2808 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.7882 \\ 0.6154 \end{bmatrix}$$

Similarly, the normalisation of the eigen vector $\varphi_1$ is given by

$$\frac{\phi_1}{\|\phi_1\|} = \frac{1}{\sqrt{(0.7808)^2 + 1^2}}\begin{bmatrix} 0.7808 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6154 \\ 0.7882 \end{bmatrix}$$

**Step 6** - KL transformation matrix from the eigen vector of the covariance matrix

From the normalised eigen vector, we have to form the transformation matrix.

$$T = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}$$

$$TT^{T} = TT^{-1} = I$$

$$TT^{T} = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}\begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix} = \begin{bmatrix} 0.9999 & 0 \\ 0 & 0.9999 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

**Step 7** - KL transformation of the input matrix

To find the KL transform of the input matrix, the formula used is Y = T[x].

$$Y_0 = T[x_0] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}\begin{bmatrix} 4 \\ -1 \end{bmatrix} = \begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix}$$

$$Y_1 = T[x_1] = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}\begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3.4226 \\ 1.1338 \end{bmatrix}$$

The final transform matrix

$$Y = \begin{bmatrix} -3.7682 & 3.4226 \\ 1.6734 & 1.1338 \end{bmatrix}.$$

Step 8 - Reconstruction of input values from the transformed coefficients

From the transform matrix, we have to reconstruct value of the given sample matrix X using the formula $X = T^{T}Y$.

$$x_0 = T^{T} Y_0 = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}\begin{bmatrix} -3.7682 \\ 1.6734 \end{bmatrix} = \begin{bmatrix} 3.9998 \\ -1 \end{bmatrix}$$

$$x_1 = T^{T} Y_1 = \begin{bmatrix} -0.7882 & 0.6154 \\ 0.6154 & 0.7882 \end{bmatrix}\begin{bmatrix} 3.4226 \\ 1.1338 \end{bmatrix} = \begin{bmatrix} -1.9999 \\ 2.9999 \end{bmatrix}$$

$$X = [x_0 \quad x_1] = \begin{bmatrix} 4 & -2 \\ -1 & 3 \end{bmatrix}.$$

## 5.5 SUMMARY

Different transform-based compression approaches have been tested with and compared to find a viable image transformation methodology for medical images of various sizes and modalities.

Image classification is a complicated process that relies on several factors. Some of the presented solutions, difficulties and more picture order potential are discussed here. The focus should be on cutting-edge classification algorithms for improving characterization precision.

## 5.6 REFERENCES

1] S. Jayaraman Digital Image Processing TMH (McGraw Hill) publication, ISBN- 13:978-0-07- 0144798

2] John G. Proakis, Digital Signal Processing: Principles, Algorithms, And Applications, 4/E

3] Gonzalez, Woods & Steven, Digital Image Processing using MATLAB, Pearson Education, ISBN-13:978-0130085191

4] https://www.robots.ox.ac.uk/~sjrob/Teaching/SP/l7.pdf

## 5.7 UNIT END EXERCISES

**1.** Compute the discrete cosine transform (DCT) matrix for N = 4.

2. Generate one Haar Basis for N = 2.

3. Compute the Haar basis for N = 8.

**4.** Compute the basis of the KL transform for the input data x1 =, $(4, 4, 5)^T$, x2 = $(3, 2, 5)^T$, x3 = $(5, 7, 6)^T$ and x4 = $(6, 7, 7)^T$.

5. Compute the 2D DFT of the 4 × 4 grayscale image given below.

$$f[m, n] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

6. . State and prove separability property of 2D-DFT.

7. Let $(x, y)$ denote a digital image of size 256 × 256. In order to compress this image, we take its Discrete Cosine Transform $(u, v)$, $u$, $v$ = 0, … ,255 and keep only the Discrete Cosine Transform coefficients for $u$, $v$ = 0, … , $n$ with $0 \leq n < 255$. The percentage of total energy of the original image that is preserved in that case is given by the formula $an + b$ + 85 with $a$, $b$ constants. Furthermore, the energy that is preserved if $n = 0$ is 85%. Find the constants $a$, $b$.

8. Image transforms are needed for _______________________________________________ .

(a) conversion information form spatial to frequency

(b) spatial domain

(c) time domain

(d) both b & c

**Answer : a**

9. The walsh and hadamard transforms are ___________in nature

(a) sinusoidal

(b) cosine

(c) non-sinusoidal

(d) cosine and sine

**Answer : c**

10. Unsampling is a process of ____________the spatial resolution of the image

(a) decreasing

(b) increasing

(c) averaging

(d) doubling

**Answer : b**

❖❖❖❖

# Module IV

# Image Restoration and Image Segmentation:

# 6

# IMAGE DEGRADATION

**Unit Structure**



## 6.0 IMAGE DEGRADATION

Image degradation is the deterioration of image quality for a variety of reasons. Image degradation occurs when the information stored with a particular image is lost by either digitization or conversion (that is, algorithmic manipulation), resulting in poor visual quality.

**Image degradation model**

The operator H acts on the input image f (x, y) with an additive noise term to model the image degradation when the degraded image g (x, y) is generated. The purpose of the restore is to get an estimate of the original image f` (x, y) and it should be as close as possible to the original image f (x, y). The degraded image is given in the spatial domain by

$$\mathbf{g(x, y) = (h \star f)(x, y) + \eta(x, y)}$$

Where

- $\eta$ (x, y) is the spatial representation of the degradation function.
- "★" indicates convolution.

Frequency domain is $\mathbf{G(u,v) = H(u,v)F(u,v) + N(u,v)}$.

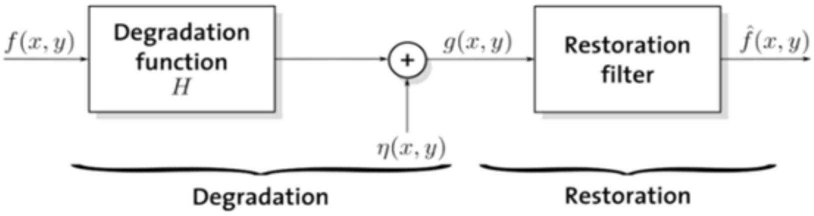



**Fig 1: Image Degradation Model**

## 6.1 CLASSIFICATION OF IMAGE RESTORATION TECHNIQUES

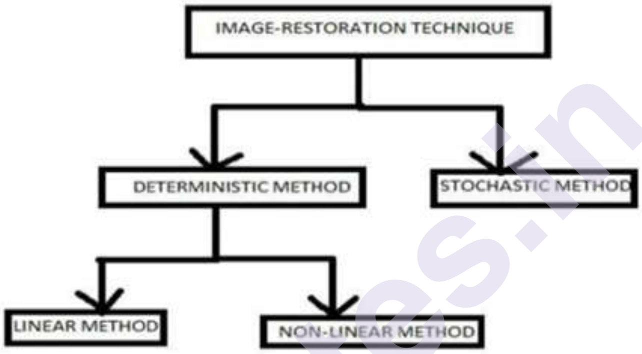



**Fig 2: Classification of Image Restoration Technique**

- ❖ **Deterministic Method: -** Prior knowledge about degradation is known.
- ❖ **Stochastic Method: -** Prior knowledge about degradation is not known.

## 6.2 IMAGE RESTORATION MODEL

Is the process of recovering an image that has been degraded by some knowledge of degradation function H and the additive noise term η(x, y).

Restoration is a process where degradation is modeled and its inverse process is applied to recover the original image.

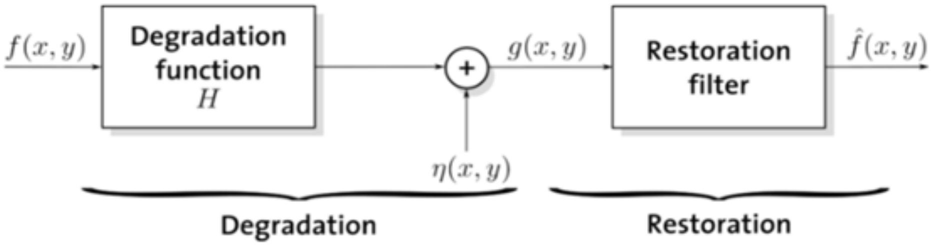



**Fig 3: Image Restoration Model**

## 6.3 IMAGE BLUR

This is the process of smoothing an image with no visible edges. If all edges are clearly visible, the image will look sharper and more detailed.

**Example 1:** Image with a face. If you can see the eyes, ears, nose, lips, forehead, etc. very clearly, you can see them clearly. This shape of the object is due to its edges. Therefore, when blurring, you reduce the edge content and make the transition from one color to another very smooth. The filter used for blurring is also called a "lowpass" filter because it allows low frequencies to penetrate and stop at high frequencies. Here, frequency means the change in pixel value. Blurred images are smooth, so the pixel values at the edges change rapidly. Therefore, it is necessary to exclude high frequencies. Filters are used for blurring purposes. For blurred images, the value of each call is 1 because the pixel values should be close to adjacent values. The filter divides by 9 for normalization. If not, the pixel value will increase and the contrast will increase, but this is not the goal.

$$Blur = 1/9\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

## 6.4 NOISE MODEL

### 6.4.1 Exponential

Exponential noise is a model where we can use to simulate data corruption. The most common reasons for it are low grade equipment and environment conditions.

Example: Photos/Images captured through an old camera end up corrupted due to lightning, temperature changes and impacts the sensors.

The PDF of *exponential* noise is given by: -

$$p(z) = \begin{cases} ae^{-z}, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

where $a = 0$. The mean and variance of $z$ are

$$\bar{z} = \frac{1}{a}$$

Exponential noise is a special case of gamma or Erlang noise where $b$ parameters equal to 1.

$$\sigma^2 = \frac{1}{a^2}$$

### 6.4.2 Uniform

The uniform noise cause by quantizing the pixels of image to a number of distinct levels is known as quantization noise, the level of the gray values of the noise are uniformly distributed across a specified range. It can be used to generate any different type of noise distribution.

**The PDF of *uniform* noise is: -**

$$p(z) = \begin{cases} \dfrac{1}{b-a}, & a \leq z \leq b \\ 0, & \text{Otherwise} \end{cases}$$

The mean and variance of $z$ are

$$\bar{z} = \frac{a+b}{2}$$

and

$$\sigma^2 = \frac{(b-a)^2}{12}$$

### 6.4.3 Salt and Pepper

Is known as impulse noise and can be caused by sharp and sudden disturbances in the image signal. This form of noise is caused due to errors in data transfer.

The PDF of *salt-and-pepper* noise is given by: -

$$p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1-(P_s + P_p) & \text{for } z = V \end{cases}$$

$k$ - Number of bits used to represent the intensity values

Range of intensity values is $[0, 2^k-1]$

❖❖❖❖

# 7

# IMAGE RESTORATION TECHNIQUES

**Unit Structure**





## 7.1 IMAGE RESTORATION TECHNIQUES

### 7.1.1 Inverse filtering

It is the process of receiving the input of a system from its output and is the simplest approach to restore the original image as the degradation function is known. The simplest approach to restoration is direct inverse filtering, where we compute an estimate, $\hat{F}(u,v)$, of the transform of the original image by dividing the transform of the degraded image, $G(u,v)$, by the degradation transfer function:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

### 7.1.2 Average filtering

Is a method of 'smoothing' images by reducing the amount of intensity variation between neighboring pixels.

Types:

- ❖ Arithmetic Mean Filter
- ❖ Geometric Mean Filter
- ❖ Harmonic Mean Filter
- ❖ Contraharmonic Mean Filter.

### 7.1.3 Median filtering

It replaces the value of a pixel by the median of the intensity levels in a predefined neighborhood of that pixel:

$$\hat{F}(x,y) = \underset{(r,c) \in S_{xy}}{\text{median}}\{g(r,c)\}$$

where

$S_{xy}$ is a subimage centered on point $(x, y)$.

## 7.2 THE DETECTION OF DISCONTINUITIES

The partitions or sub-division of an image is based on some abrupt changes in the intensity level of images and is used for detecting three basic types of grey-level discontinuities in a digital image: Points, Lines and Edges. To identify these, 3* 3 mask operation is used.

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

The response of the mask at any point in the image is given by: -

$$R = w_1z_1 + w_2z_2 + \cdots w_9z_9$$
$$= \sum_{i=1}^{9} w_i z_i$$

where

$z_i$ is gray-level of pixel associated with mask coefficient $w_i$.

### 7.2.1 Point detection

A point is the basic type of discontinuity in a digital image. The most common way to finding discontinuities is to run a $(n * n)$ mask over each point in the image. The detection of isolated point different from constant background image can be done using the following mask:

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

The point is detected at a location (x, y) in an image where the mask is centered. If the corresponding value of R such that:

$$|R| \geq T$$

Where R is the response of the mask at any point in the image and T is non-negative threshold value. It means that isolated point is detected at the corresponding value (x, y).

The result of point detection mask is shown in Fig 4:

**Fig 4: Point Detection**

**and**

### 7.2.2 Line detection

It is the process of receiving the input of a system from its output and is the simplest approach to restore the original image as the degradation function is known. The simplest approach to restoration is direct inverse filtering, where we compute an estimate, F □(u,v), of the transform of the original image by dividing the transform of the degraded image, G(u,v), by the degradation transfer function:

Line detection is the level of complexity in the direction of image discontinuity. Consider the mask shown in masks. If the first mask were moved around an image, it would respond more strongly to lines (one pixel thick) oriented horizontally. With a constant background, the maximum response would result when the line passed through the middle row of the mask and can be easily verified by sketching a simple array of 1`s with a line of a different gray level (say, 5`s) running horizontally through the array.

Suppose R1, R2, R3, and R4 represent the mask response of the specific mask below from left to right. Where R is given by:

$$R = w_1z_1 + w_2z_2 + \cdots w_9z_9$$

Suppose that the four masks are run individually through an image. If, at a certain point in the image, $|R_i| > |R_j|$, for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

**(a) Horizontal**

| | | |
|---|---|---|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

**(b) +45**

| | | |
|---|---|---|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

**(c) Vertical**

| | | |
|---|---|---|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

**(d) -45**

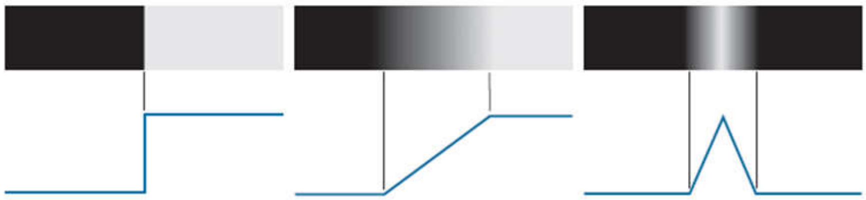
### 7.2.3 Edge detections

Significant transitions in an image are called as edges.

Types of edges

- ❖ Horizontal edges
- ❖ Vertical Edges
- ❖ Diagonal Edges

Edge detection is the most common approach to detecting something meaningful. Grayscale discontinuity. An edge is the boundary between two regions with different intensity levels. In practice, the edges of a digital image are blurry and noisy, the degree of blurring is primarily determined by the limitations of the focusing mechanism (such as the lens in the case of optical images), and the noise level is primarily determined by the electronic components of the imaging system. Will be decided. .. In such situations, the edges are modeled closer, as if they had a slanted profile. The tilt of the ramp is inversely proportional to the degree of blurring of the edges. In this model, there is no single "edge point" along the profile. Instead, an edge point now is any point contained in the ramp, and an edge segment would then be a set of such points that are connected. A third type of edge is the so-called *roof edge*, having the characteristics illustrated in Fig below. Roof edges are models of lines through a region, with the base (width) of the edge being determined by the thickness and sharpness of the line.



From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

**Fig 5: Edge Detection**

## 7.3 VARIOUS METHODS USED FOR EDGE DETECTION

### Detection of edges

Most of the shape information of an image is enclosed in edges. So first we detect these edges in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer.

- ❖ Prewitt Operator
- ❖ Sobel Operator
- ❖ Robinson Compass Masks
- ❖ Krisch Compass Masks
- ❖ Laplacian Operator.

All the filters mentioned above are Linear filters.

### 7.3.1 Prewitt Filter or Prewitt Operator

It is used for edge detection in an image detecting both types of edges.

- ❖ Horizontal edges or along the x-axis.
- ❖ Vertical Edges or along the y-axis.

*Prewitt Operator [X-axis] = [ -1 0 1; -1 0 1; -1 0 1]*

*Prewitt Operator [Y-axis] = [-1 -1 -1; 0 0 0; 1 1 1]*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

**Horizontal Direction**

**Fig 6: Horizontal Direction**

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

**Vertical Direction**

**Fig 7: Vertical Direction**

### 7.3.2 Sobel Filter or Sobel Operator

Sobel Filter looks similar to Prewitt operator; it is a derivate mask used for edge detection. Sobel operator is also used to detect two kinds of edges in an image:

- ❖ Horizontal direction.
- ❖ Vertical direction.

Major difference is that in sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

This mask works exactly same as the Prewitt operator vertical mask. The only one difference it has "2" and "-2" values in center of first and third column. As applied on an image this mask will highlight the vertical edges.

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

Horizontal Direction

**Fig 8: Horizontal Direction**

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

Vertical Direction

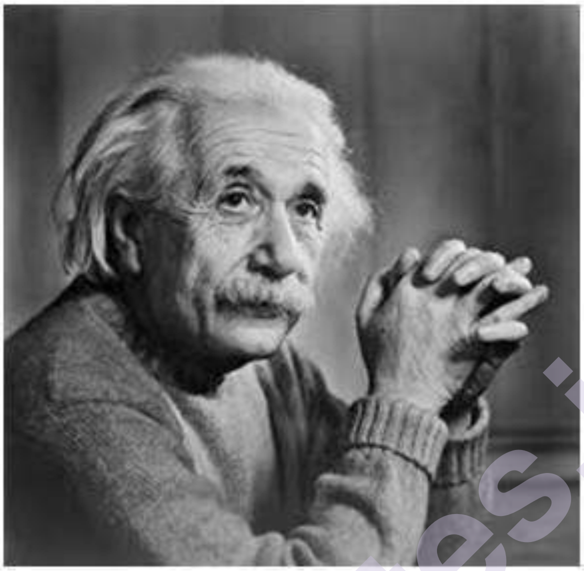
**Fig 9: Vertical Direction**

How it works

This mask enhances the horizontal edges of the image. It also works on the basis of the mask principle above to calculate the difference in pixel intensity for a particular edge. The center mask row consists of zeros, so it does not contain the original edge values of the image, but it does calculate the difference in pixel intensity above and below each edge. This amplifies the sudden changes in intensity and makes the edges easier to see. Let's see these masks in action:

**Sample Image**

Following is a sample picture on which we will apply above two masks one at time.



**After applying Vertical Mask**

After applying vertical mask on the above sample image, following image will be obtained.

**After applying Horizontal Mask**

After applying horizontal mask on the above sample image, following image will be obtained

**Comparison**

As you can see, in the first image to which the vertical mask is applied, all vertical edges are easier to see than the original image. Similarly, in the second image, all horizontal edges are shown as a result of applying the horizontal mask.

In this way, you can see that both horizontal and vertical edges of the image can be detected. Also, if you compare the result of the Sobel operator with the Prewitt operator, you can see that the Sobel operator finds more edges and makes the edges easier to see than the Prewitt operator.

This is because the Sobel operator gave more weight to the pixel weight of the edges.

**Applying more weight to mask**

Applying more weight to the mask, the more edges it will get for us.

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

Compare the result of this mask with of the Prewitt vertical mask, it is apparent that this mask will give out more edges as compared to Prewitt one just because we have allotted more weight in the mask.

### 7.3.3 Fri-Chen Filter Hough Transform

Fri-Chen edge detector is also a first order operation Prewitt and Sobel operator. Frei-Chen masks are unique masks, contains all the basis vectors. This means that a 3×3 image area is represented with the weighted sum of nine Frei-Chen masks that can be seen below: -

$$G_1 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \qquad G_2 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_3 = \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \qquad G_4 = \frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

$$G_5 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$G_6 = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \qquad G_7 = \frac{1}{6}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$G_8 = \frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} \qquad G_9 = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

## 7.4 THRESHOLDING REGION BASED SEGMENTATION CHAIN CODES

Pixels are categorized based on the range of values they contain. The figure below shows the boundaries obtained by thresholding the muscle fiber image. Pixel values less than 128 were placed in one category and the rest in the other.

**7.4.1 Region-based segmentation** algorithms algorithm works repeatedly to group adjacent pixels with similar values and split groups of pixels with different values.

### 7.4.2 Region-based segmentation Chain codes

Boundary represented by a connected sequence of staraight-line segments of specified length and direction(4 or 8 connectivity).

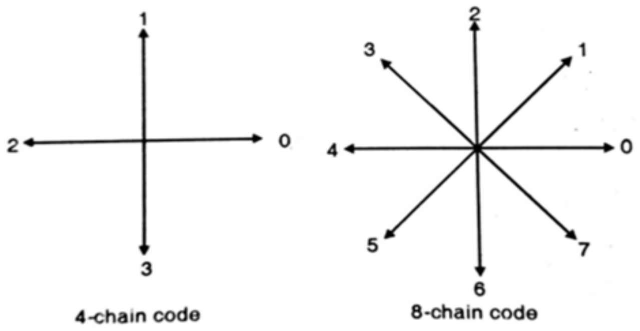
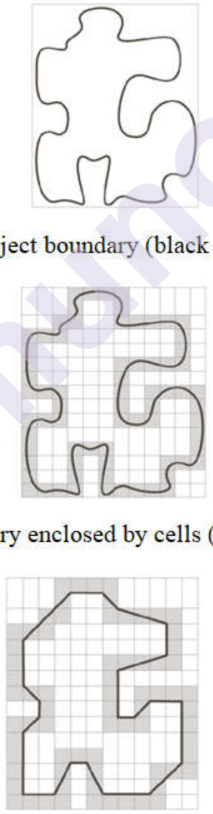



**Fig 10: Region based Segmentation Chain Codes**

## 7.5 POLYGON APPROXIMATION

Polygon approximation is used to represent boundaries in straight lines, and closed paths are polygons. The number of straight line segments used determines the accuracy of the approximation. You need to use the minimum number of sides needed to hold the required shape information (minimum perimeter polygons). A large number of edges only adds noise to the model. Polygon approximation using minimum perimeter polygons:



(a) An object boundary (black curve)

(b) Boundary enclosed by cells (in gray)

(c) Minimum perimeter obtained by allowing the boundary to shrink

**Fig 11: Polygon approximation**

### 7.5.1 Shape numbers

As shown in the figure below, the shape number of the Freeman chain-coded boundary based on the 4-way code is defined as the first difference in minimum magnitude. The order n, of a shape is defined as the number of digits in the representation. Moreover, for closed boundaries, n is even, and its value limits the number of different shapes possible. The first difference in the 4-way directional chain code is independent of rotation (in 90 ° increments), but the coded boundaries usually depend on the on the orientation of the grid.

Depending on how the grid spacing is selected, the resulting shape number order is usually equal to n, but borders with indentations comparable to this spacing may produce shape numbers greater than n. In this case, specify a rectangle with an order less than n and repeat the process until the resulting shape number is nth. The order of form numbers starts at 4, and we are using 4 connections, so we always need it.

The border is closed.

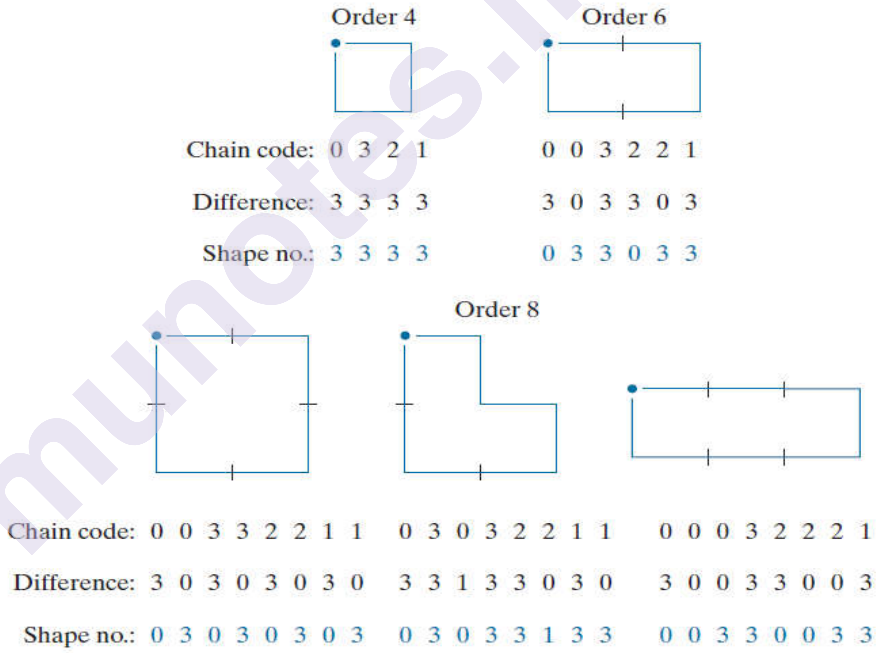



**Fig 12: Shape Numbers**

## 7.6 REFERENCES


1. Pratt WK. Introduction to digital image processing. CRC press; 2013 Sep 13.
2. Niblack W. An introduction to digital image processing. Strandberg Publishing Company; 1985 Oct 1.
3. Burger W, Burge MJ, Burge MJ, Burge MJ. Principles of digital image processing. London: Springer; 2009.

4. Jain AK. Fundamentals of digital image processing. Prentice-Hall, Inc.; 1989 Jan 1.
5. Dougherty ER. Digital image processing methods. CRC Press; 2020 Aug 26.
6. Gonzalez RC. Digital image processing. Pearson education india; 2009.
7. Marchand-Maillet S, Sharaiha YM. Binary digital image processing: a discrete approach. Elsevier; 1999 Dec 1.
8. Andrews HC, Hunt BR. Digital image restoration.
9. Lagendijk RL, Biemond J. Basic methods for image restoration and identification. InThe essential guide to image processing 2009 Jan 1 (pp. 323-348). Academic Press.
10. Banham MR, Katsaggelos AK. Digital image restoration. IEEE signal processing magazine. 1997 Mar;14(2):24-41.
11. Hunt BR. Bauesian Methods in Nonkinear Digital Image Restoration. IEEE Transactions on Computers. 1977 Mar 1;26(3):219-29.
12. Figueiredo MA, Nowak RD. An EM algorithm for wavelet-based image restoration. IEEE Transactions on Image Processing. 2003 Aug 4;12(8):906-16.
13. Digital Image Processing – Tutorialspoint. https://www.tutorialspoint.com/dip/index.htm.
14. Types of Restoration Filters. https://www.geeksforgeeks.org/types-of-restoration-filters/.

## 7.7 MOOCS

1. Digital Image Processing. https://onlinecourses.nptel.ac.in/noc19_ee55/preview.
2. Digital Image Processing. https://www.mygreatlearning.com/academy/learn-for-free/courses/digital-image-processing.
3. Fundamentals of Digital Image Processing. https://alison.com/course/fundamentals-of-digital-image-processing
4. Digital Image Processing: Operations and Applications. https://www.udemy.com/course/digital-image-processing-operations-and-applications/

5. Digital Image Processing. https://www.udemy.com/course/digital-image-processing-made-easy/.

## 7.8 VIDEO LINKS

1. Digital Image Processing.
   https://www.youtube.com/watch?v=sa7vO6YXBik&list=PL3rE2jS8zxAykFjinlf6EsucLv5EA03_m.
2. Digital Image Processing - Introduction of DIP.
   https://www.youtube.com/watch?v=iZmHHVwp0Ow&list=PL3rE2jS8zxAykFjinlf6EsucLv5EA03_m&index=2.
3. Digital Image Processing - Nature of Image Processing & Applications.
   https://www.youtube.com/watch?v=UqKQ_lfDwx8&list=PL3rE2jS8zxAykFjinlf6EsucLv5EA03_m&index=3.
4. Digital Image Processing - Image Smoothing Spatial Filters.
   https://www.youtube.com/watch?v=Dtdmm7QodO4&list=PL3rE2jS8zxAykFjinlf6EsucLv5EA03_m&index=31.
5. Digital Image Processing - Image Degradation (Restoration) Model.
   https://www.youtube.com/watch?v=U1h0biwb8OM&t=23s.
6. Digital Image Processing - Estimation of Degradation Function.
   https://www.youtube.com/watch?v=n5dlO82SwJU.
7. Image Restoration: Estimation of Degradation Function.
   https://www.youtube.com/watch?v=fkgxpXx0250.
8. Estimating the Degradation function in Digital Image Processing | Observation | Experimentation | Modeling.
   https://www.youtube.com/watch?v=cloLOHb5F_k.
9. Digital Image Processing - Image Restoration Techniques.
   https://www.youtube.com/watch?v=PBhBw5qfaq4.
10. Estimation of Degradation Model and Restoration Techniques – I.
    https://www.youtube.com/watch?v=3XQcZeNF_8k
11. Image Degradation and Restoration and Model of Image Degradation and Restoration process in DIP.
    https://www.youtube.com/watch?v=w0YNkSQxvwo.

12. Image Restoration Techniques – I.
https://www.youtube.com/watch?v=MrNafUqh860.
13. Image degradation and restoration | Digital Image Processing.
https://www.youtube.com/watch?v=ScBBAHHxepY.
14. Degradation function.
https://www.youtube.com/watch?v=dIC53nDnwgk.

## 7.9 QUIZ

1. What is Digital Image Processing?
a) It's an application that alters digital videos
b) It's a software that allows altering digital pictures
c) It's a system that manipulates digital medias
d) It's a machine that allows altering digital images
**ANSWER: B**

2. Which of the following process helps in Image enhancement?
a) Digital Image Processing
b) Analog Image Processing
c) Both a and b
d) None of the above
**ANSWER: C**

3. Among the following, functions that can be performed by digital image processing is?
a) Fast image storage and retrieval
b) Controlled viewing
c) Image reformatting
d) All of the above
**ANSWER: D**

4. Which of the following is an example of Digital Image Processing?
a) Computer Graphics
b) Pixels
c) Camera Mechanism
d) All of the mentioned
**ANSWER: D**

5. What are the categories of digital image processing?
a) Image Enhancement
b) Image Classification and Analysis
c) Image Transformation
d) All of the mentioned
**ANSWER: D**

6. How does picture formation in the eye vary from image formation in a camera?
a) Fixed focal length
b) Varying distance between lens and imaging plane
c) No difference
d) Variable focal length
**ANSWER: D**

7. What are the names of the various colour image processing categories?
a) Pseudo-color and Multi-color processing
b) Half-color and pseudo-color processing
c) Full-color and pseudo-color processing
d) Half-color and full-color processing
**ANSWER: C**

8. Which characteristics are taken together in chromaticity?
a) Hue and Saturation
b) Hue and Brightness
c) Saturation, Hue, and Brightness
d) Saturation and Brightness
**ANSWER: A**

9. Which of the following statement describe the term pixel depth?
a) It is the number of units used to represent each pixel in RGB space
b) It is the number of mm used to represent each pixel in RGB space
c) It is the number of bytes used to represent each pixel in RGB space
d) It is the number of bits used to represent each pixel in RGB space
**ANSWER: D**

10. The aliasing effect on an image can be reduced using which of the following methods?
a) By reducing the high-frequency components of image by clarifying the image
b) By increasing the high-frequency components of image by clarifying the image

c) By increasing the high-frequency components of image by blurring the image

d) By reducing the high-frequency components of image by blurring the image

**ANSWER: D**

11. Which of the following is the first and foremost step in Image Processing?

a) Image acquisition

b) Segmentation

c) Image enhancement

d) Image restoration

**ANSWER: A**

12. Which of the following image processing approaches is the fastest, most accurate, and flexible?

a) Photographic

b) Electronic

c) Digital

d) Optical

**ANSWER: C**

13. Which of the following is the next step in image processing after compression?

a) Representation and description

b) Morphological processing

c) Segmentation

d) Wavelets

**ANSWER: B**

14. ___________ determines the quality of a digital image.

a) The discrete gray levels

b) The number of samples

c) discrete gray levels & number of samples

d) None of the mentioned

**ANSWER: C**

15. Image processing involves how many steps?

a) 7

b) 8

c) 13

d) 10

**ANSWER: D**

16. Which of the following is the abbreviation of JPEG?

a) Joint Photographic Experts Group

b) Joint Photographs Expansion Group

c) Joint Photographic Expanded Group

d) Joint Photographic Expansion Group

**ANSWER: A**

17. Which of the following is the role played by segmentation in image processing?

a) Deals with property in which images are subdivided successively into smaller regions

b) Deals with partitioning an image into its constituent parts or objects

c) Deals with extracting attributes that result in some quantitative information of interest

d) Deals with techniques for reducing the storage required saving an image, or the bandwidth required transmitting it

**ANSWER: B**

18. The digitization process, in which the digital image comprises M rows and N columns, necessitates choices for M, N, and the number of grey levels per pixel, L. M and N must have which of the following values?

a) M have to be positive and N have to be negative integer

b) M have to be negative and N have to be positive integer

c) M and N have to be negative integer

d) M and N have to be positive integer

**ANSWER: D**

19. Which of the following tool is used in tasks such as zooming, shrinking, rotating, etc.?

a) Filters

b) Sampling

c) Interpolation

d) None of the Mentioned

**ANSWER: C**

20. The effect caused by the use of an insufficient number of intensity levels in smooth areas of a digital image _____________

a) False Contouring

b) Interpolation

c) Gaussian smooth

d) Contouring

**ANSWER: A**

21. What is the procedure done on a digital image to alter the values of its individual pixels known as?

a) Geometric Spacial Transformation

b) Single Pixel Operation

c) Image Registration

d) Neighbourhood Operations

**ANSWER: B**

22. Points whose locations are known exactly in the input and reference images are used in Geometric Spacial Transformation.

a) Known points

b) Key-points

c) Réseau points

d) Tie points

**ANSWER: D**

23. ___________ is a commercial use of Image Subtraction.

a) MRI scan

b) CT scan

c) Mask mode radiography

d) none of the mentioned

**ANSWER: C**

24. Approaches to image processing that work directly on the pixels of incoming image work in ____________

a) Spatial domain

b) Inverse transformation

c) Transform domain

d) None of the Mentioned

**ANSWER: A**

25. Which of the following in an image can be removed by using a smoothing filter?

a) Sharp transitions of brightness levels

b) Sharp transitions of gray levels

c) Smooth transitions of gray levels

d) Smooth transitions of brightness levels

**ANSWER: B**

26. Region of Interest (ROI) operations is generally known as _______

a) Masking

b) Dilation

c) Shading correction

d) None of the Mentioned

**ANSWER: A**

27. Which of the following comes under the application of image blurring?

a) Image segmentation

b) Object motion

c) Object detection

d) Gross representation

**ANSWER: D**

28. Which of the following filter's responses is based on the pixels ranking?

a) Sharpening filters

b) Nonlinear smoothing filters

c) Geometric mean filter

d) Linear smoothing filters

**ANSWER: B**

29. Which of the following illustrates three main types of image enhancing functions?

a) Linear, logarithmic and power law

b) Linear, logarithmic and inverse law

c) Linear, exponential and inverse law

d) Power law, logarithmic and inverse law

**ANSWER: D**

30. Which of the following is the primary objective of sharpening of an image?

a) Decrease the brightness of the image

b) Increase the brightness of the image

c) Highlight fine details in the image

d) Blurring the image

**ANSWER: C**

31. Which of the following operation is done on the pixels in sharpening the image, in the spatial domain?

a) Differentiation

b) Median

c) Integration

d) Average

**ANSWER: A**

32. ________ is the principle objective of Sharpening, to highlight transitions.

a) Brightness

b) Pixel density

c) Composure

d) Intensity

**ANSWER: D**

33. _________ enhance Image Differentiation?

a) Pixel Density

b) Contours

c) Edges

d) None of the mentioned

**ANSWER: C**

34. Which of the following fact is correct for an image?

a) An image is the multiplication of illumination and reflectance component

b) An image is the subtraction of reflectance component from illumination component

c) An image is the subtraction of illumination component from reflectance component

d) An image is the addition of illumination and reflectance component

**ANSWER: A**

35. Which of the following occurs in Unsharp Masking?

a) Subtracting blurred image from original

b) Blurring the original image

c) Adding a mask to the original image

d) All of the mentioned

**ANSWER: D**

36. Which of the following makes an image difficult to enhance?

a) Dynamic range of intensity levels

b) High noise

c) Narrow range of intensity levels

d) All of the mentioned

**ANSWER: D**

37. _________ is the process of moving a filter mask over the image and computing the sum of products at each location.

a) Nonlinear spatial filtering

b) Convolution

c) Correlation

d) Linear spatial filtering

**ANSWER: C**

38. Which side of the greyscale is the components of the histogram concentrated in a dark image?

a) Medium

b) Low

c) Evenly distributed

d) High

**ANSWER: B**

39. Which of the following is the application of Histogram Equalisation?

a) Blurring

b) Contrast adjustment

c) Image enhancement

d) None of the Mentioned

**ANSWER: C**

40. Which of the following is the expansion of PDF, in uniform PDF?

a) Probability Density Function

b) Previously Derived Function

c) Post Derivation Function

d) Portable Document Format

**ANSWER: A**

41. ____________ filter is known as averaging filters.

a) Bandpass

b) Low pass

c) High pass

d) None of the Mentioned

**ANSWER: B**

42. What is/are the resultant image of a smoothing filter?

a) Image with reduced sharp transitions in gray levels

b) Image with high sharp transitions in gray levels

c) None of the mentioned

d) All of the mentioned

**ANSWER: A**

43. The response for linear spatial filtering is given by the relationship __________

a) Difference of filter coefficient's product and corresponding image pixel under filter mask

b) Product of filter coefficient's product and corresponding image pixel under filter mask

c) Sum of filter coefficient's product and corresponding image pixel under filter mask

d) None of the mentioned

**ANSWER: C**

44. ___________ is/are the feature(s) of a highpass filtered image.

a) An overall sharper image

b) Have less gray-level variation in smooth areas

c) Emphasized transitional gray-level details

d) All of the mentioned

**ANSWER: D**

45. The filter order of a Butterworth lowpass filter determines whether it is a very sharp or extremely smooth filter function, or an intermediate filter function. Which of the following filters does the filter approach if the parameter value is very high?

a) Gaussian lowpass filter

b) Ideal lowpass filter

c) Gaussian & Ideal lowpass filters

d) None of the mentioned

**ANSWER: B**

46. Which of the following image component is characterized by a slow spatial variation?

a) Reflectance and Illumination components

b) Reflectance component

c) Illumination component

d) None of the mentioned

**ANSWER: C**

47. Gamma Correction is defined as __________

a) Light brightness variation

b) A Power-law response phenomenon

c) Inverted Intensity curve

d) None of the Mentioned

**ANSWER: B**

48. ____________________ is known as the highlighting the contribution made to total image by specific bits instead of highlighting intensity-level changes.

a) Bit-plane slicing

b) Intensity Highlighting

c) Byte-Slicing

d) None of the Mentioned

**ANSWER: A**

49. Which gray-level transformation increases the dynamic range of gray-level in the image?

a) Negative transformations

b) Contrast stretching

c) Power-law transformations

d) None of the mentioned

**ANSWER: B**

50. What is/are the gray-level slicing approach(es)?

a) To brighten the pixels gray-value of interest and preserve the background

b) To give all gray level of a specific range high value and a low value to all other gray levels

c) All of the mentioned

d) None of the mentioned

**ANSWER: C**

❖❖❖❖

# Module V

# 8

# IMAGE DATA COMPRESSION AND MORPHOLOGICAL OPERATION

**Unit Structure**



## 8.1 NEED FOR COMPRESSION

Image compression is one of the most important and commercially successful technologies in the field of digital image processing. It involves the art and science of minimizing the amount of data required to represent an image. Image compression is a technique for reducing the amount of data needed to represent a digital image. It's crucial for lowering storage requirements and increasing transmission speeds.

It aims to decrease the irrelevance and redundancy of image data in order to store or transmit data more efficiently. Its goal is to reduce the amount of bits needed to represent an image.

Consider a black-and-white image with a resolution of 1000*1000 pixels and an intensity of 8 bits per pixel. So total number of bits required per image is 1000*1000*8 = 80,00,000 bits. Consider the total bits for a video of 3 seconds with 30 frames per second of the above-mentioned kind images: 3*(30*(8, 000, 000))=720, 000, 000 bits.

As we've seen, just storing a 3-second video requires a large number of bits. As a result, we need a technique to have a suitable representation as well as a way to retain image information in a small number of bits without affecting the image's character. As a result, image compression is crucial.

## 8.2 REDUNDANCY IN IMAGE

Redundancy refers to "storing additional information to represent a set of information." We know that computers store images in pixel values. Therefore, the pixel values of the image may be duplicated, or even if some pixel values are deleted, the information in the actual image may not be affected. 3-Types of Image redundancy: -

**a)** ***Coding redundancy*: -**

The symbols such as letters, numbers, bits, and so on are used to represent a set of data or events and collection of these symbols is known as code. Each code word's length is determined by the number of symbols it contains. In most 2-D intensity arrays, the 8-bit codes used to represent the intensities contain more bits than are required to express the intensities.

**b) Spatial and temporal redundancy: -**

Because most 2-D intensity array pixels are spatially interconnected (i.e., each pixel is similar to or dependent on surrounding pixels), information is duplicated in the representations of the correlated pixels unnecessarily. Temporally interconnected pixels (those that are similar to or dependent on pixels in surrounding frames) in a video series also duplicate information.

**c) Irrelevant information: -**

Human visual system ignore most of the 2-D intensity arrays that contain data. If that data is not used it is considered to be as redundant.

## 8.3 CLASSIFICATION OF IMAGE COMPRESSION SCHEMES

Two types of Image compression technique: -

a) Lossy image compression: - Lossy compression means to reduce the image size while discarding some data from the original image file.

b) Lossless image compression: - The lossless image compression approach involves representing an image signal with the least amount of bits possible without losing any information, resulting in faster transmission and reduced storage requirement

**Further, these two techniques are also classified as follows**

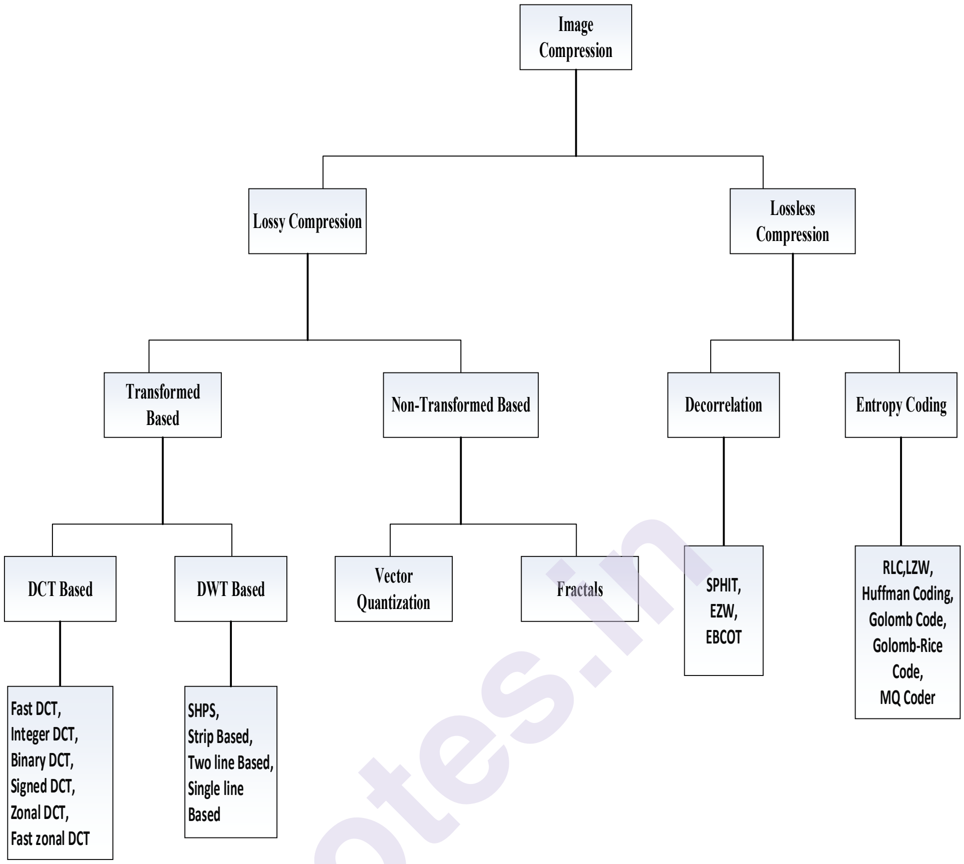



**Fig 1: Classification of Image compression schemes**

## 8.4 HUFFMAN CODING

The Huffman coding technique is a lossless image compression method. Huffman coding is based on the frequency of data item in order of their occurrences, such as a pixel in an image, appears. These codes are variable length code. It can be found in JPEG files.

**Steps and example: -**

**Forward Pass: -**

1. Sort probabilities of each symbol.
2. Combine two probabilities having lowest probability values.
3. Repeat Step2 until only two probabilities remain.

| Original source | | Source reduction | | | |
|---|---|---|---|---|---|
| Symbol | Probability | 1 | 2 | 3 | 4 |
| $a_2$ | 0.4 | 0.4 | 0.4 | 0.4 | 0.6 |
| $a_6$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.4 |
| $a_1$ | 0.1 | 0.1 | 0.2 | 0.3 | |
| $a_4$ | 0.1 | 0.1 | 0.1 | | |
| $a_3$ | 0.06 | 0.1 | | | |
| $a_5$ | 0.04 | | | | |

**Fig 2: Forward Pass**

**Backward Pass**

❖ Assign code symbols going backwards.

| Original source | | | Source reduction | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Symbol | Probability | Code | 1 | | 2 | | 3 | | 4 | |
| $a_2$ | 0.4 | 1 | 0.4 | 1 | 0.4 | 1 | 0.4 | 1 | 0.6 | 0 |
| $a_6$ | 0.3 | 00 | 0.3 | 00 | 0.3 | 00 | 0.3 | 00 | 0.4 | 1 |
| $a_1$ | 0.1 | 011 | 0.1 | 011 | 0.2 | 010 | 0.3 | 01 | | |
| $a_4$ | 0.1 | 0100 | 0.1 | 0100 | 0.1 | 011 | | | | |
| $a_3$ | 0.06 | 01010 | 0.1 | 0101 | | | | | | |
| $a_5$ | 0.04 | 01011 | | | | | | | | |

**Fig 3: Backward Pass**



**So, average length of this code: -**

$$L_{avg}=E(I(a_k))=\sum_{k=1}^{6} I(a_k)p(a_k)=(0.4)(1)+(0.3)(2)+(0.1)(3)+(0.1)(4)+(0.06)(5)+(0.04)(5)$$

$$= 2.2 \text{ bits/pixel}$$

## 8.5 ARITHMETIC CODING

Arithmetic coding is a lossless image compression technique. Arithmetic coding generates non-block. A single arithmetic code word is assigned to a complete sequence of source symbols (or message). The code word itself designates a range of actual numbers from 0 to 1. Each symbol in the message shrinks the interval in proportion to its occurrence probability.

**Steps and example: -**

Figure 1 illustrates the basic arithmetic coding process. A five-symbol sequence or message, a1a2a3a3a4, is coded here from a four-symbol source. The message is supposed to occupy the entire half-open interval [0, 1] at the start of the coding procedure. This interval is initially partitioned into four sections depending on the probabilities of each source symbol, as shown in Table below, for example, symbol a1 is related with Subinterval [0, 0.2]. The message interval is initially limited to [0, 0.2] because it is the first symbol of the message being coded.

| Source Symbol | Probability | Initial Subinterval |
|---|---|---|
| $a_1$ | 0.2 | [0.0,0.2) |
| $a_2$ | 0.2 | [0.2,0.4) |
| $a_3$ | 0.4 | [0.4,0.8) |
| $a_4$ | 0.2 | [0.8,1.0) |

The range [0, 0.2] is enlarged to the full height of the figure, with the values of the narrowed range labeling its end points. The narrower range is then subdivided according with probabilities of the original source symbol, and the process is repeated for the next message symbol. Symbols a2 and a3 narrow the subinterval to [0.04, 0.08], 0.056, 0.072, and so on. The range is narrowed to [0.06752, 0.0688) when the last message sign is used as a specific end-of-message indicator. Of fact, the message can be represented by any number within this subinterval, such as 0.068.

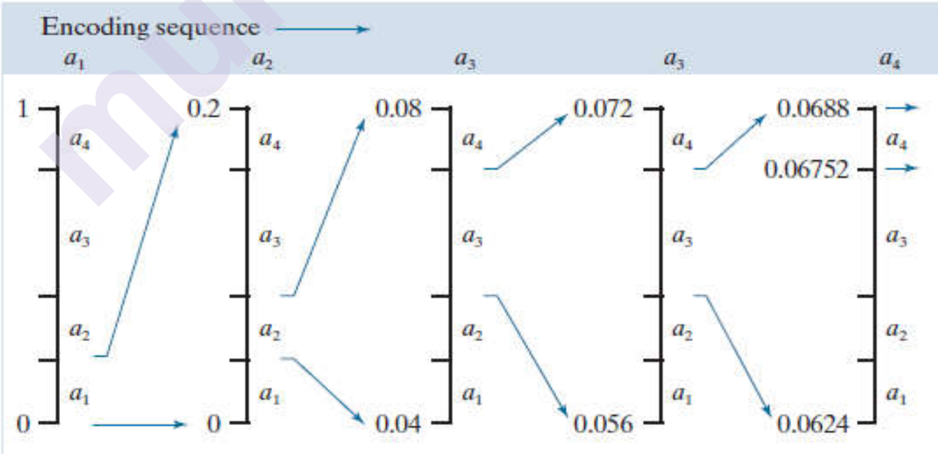



**Fig 4: Encoding Sequence**

## 8.6 DICTIONARY BASED COMPRESSION

This method is not statistically based. The characteristic of this strategy is that it is fast and adaptable. The dictionary based compression replaces input strings with a code to an entry in a dictionary. The Lempel-Ziv-Welch (LZW) algorithm is the most well-known dictionary-based approach.

## 8.7 LEMPEL-ZIV-WELCH (LZW) ALGORITHM

Lempel-Ziv-Welch (LZW) is an error-free compression approach. This technique assigns fixed-length code words to variable length sequences of source symbols. LZW coding is distinguished by the fact that it does not require prior information of the probability of occurrence of the symbols to be encoded. GIF, TIFF, and PDF are just a few of the popular image file formats that have LZW compression built in.

**For Example:**

The grey values 0, 1, 2,..., and 255 are assigned to the first 256 words in the dictionary for 8-bit monochrome images. As the encoder sequentially examines the image's pixels, gray- level sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations. If the first two pixels of the image are white, for instance, sequence —255- 255□ might be assigned to location 256, the address following the locations reserved for gray levels 0 through 255. The next time that two consecutive white pixels are encountered, code word 256, the address of the location containing sequence 255-255, is used to represent them. If a 9-bit, 512-word dictionary is employed in the coding process, the original (8 + 8) bits that were used to represent the two pixels are replaced by a single 9-bit code word.

Consider the following 4 x 4, 8-bit image of a vertical edge: -

| | | | |
|---|---|---|---|
| 39 | 39 | 126 | 126 |
| 39 | 39 | 126 | 126 |
| 39 | 39 | 126 | 126 |
| 39 | 39 | 126 | 126 |



Figure 5 details the steps involved in coding its 16 pixels. A 512-word dictionary with the following starting content is assumed:

| Dictionary Location | Entry |
|---|---|
| 0 | 0 |
| 1 | 1 |
| ⋮ | ⋮ |
| 255 | 255 |
| 256 | – |
| ⋮ | ⋮ |
| 511 | – |

**Fig 5: A 512-word dictionary**

Locations 256 through 511 are initially unused. The image is encoded by processing its pixels in a left-to-right, top-to-bottom manner. Each successive gray-level value is concatenated with a variable—column 1 of Figure 6 called the "currently recognized sequence." As can be seen, this variable is initially null or empty. The dictionary is searched for each concatenated sequence and if found, as was the case in the first row of the table, is replaced by the newly concatenated and recognized (i.e., located in the dictionary) sequence. This was done in column 1 of row 2.

| Currently Recognized Sequence | Pixel Being Processed | Encoded Output | Dictionary Location (Code Word) | Dictionary Entry |
|---|---|---|---|---|
| | 39 | | | |
| 39 | 39 | 39 | 256 | 39-39 |
| 39 | 126 | 39 | 257 | 39-126 |
| 126 | 126 | 126 | 258 | 126-126 |
| 126 | 39 | 126 | 259 | 126-39 |
| 39 | 39 | | | |
| 39-39 | 126 | 256 | 260 | 39-39-126 |
| 126 | 126 | | | |
| 126-126 | 39 | 258 | 261 | 126-126-39 |
| 39 | 39 | | | |
| 39-39 | 126 | | | |
| 39-39-126 | 126 | 260 | 262 | 39-39-126-126 |
| 126 | 39 | | | |
| 126-39 | 39 | 259 | 263 | 126-39-39 |
| 39 | 126 | | | |
| 39-126 | 126 | 257 | 264 | 39-126-126 |
| 126 | | 126 | | |

**Fig 6: Currently recognized sequence**

## 8.8 TRANSFORM BASED COMPRESSION

Transform coding is performed by taking an image and breaking it down into sub-image (block) of size nxn. The transform is then applied to each sub-image (block) and the resulting transform coefficients are quantized and entropy coded, divides an image into small non-overlapping blocks of equal size (e.g., 8 * 8) and and using 2-D transform it processes the block of image independently. To map each block of images into a set of transform coefficients the block transform coding uses linear transform. A significant number of coefficients with small magnitudes can be quantized for most images.

Typical blocks transform coding system

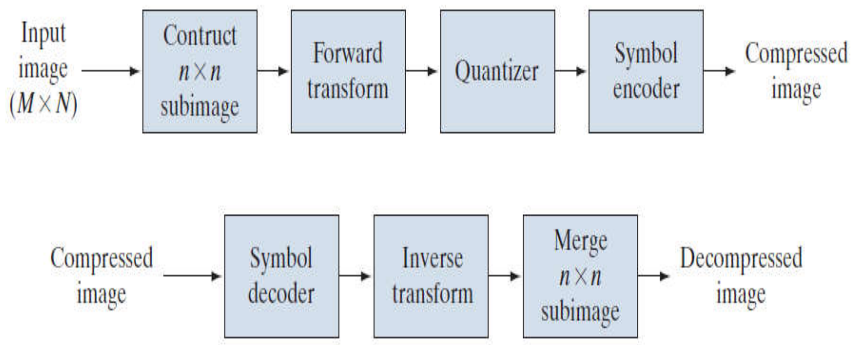



Fig 7: Typical blocks transform coding system

❖❖❖❖

# 9

# IMAGE COMPRESSION STANDARDS

**Unit Structure**



## 9.1 JPEG (JOINT PHOTOGRAPH EXPERT GROUP)

JPEG is a lossy image compression standard, which means that some details may be lost when the image is restored from the compressed data. JPEG is designed for full-color or grayscale images of natural scenes. It works very well with photographic images. JPEG does not work as well on images with sharp edges or artificial scenes such as graphical drawings, text documents, or cartoon pictures. A product or system must support the basic system in order make JPEG compatible. The precision in baseline of the input and output data is limited to 8 bits, while the quantized DCT values are limited to 11 bits. The compression method involves three steps first one DCT computation second being Quantization and final is variable-length code assignment. The image is firstly segmented into 8-bit pixel blocks, which are processed from left to right and top to bottom.

**Working of JPEG compression**

**Steps and Example: -**

First step is to divide an image into blocks with each having dimensions of 8 x8.

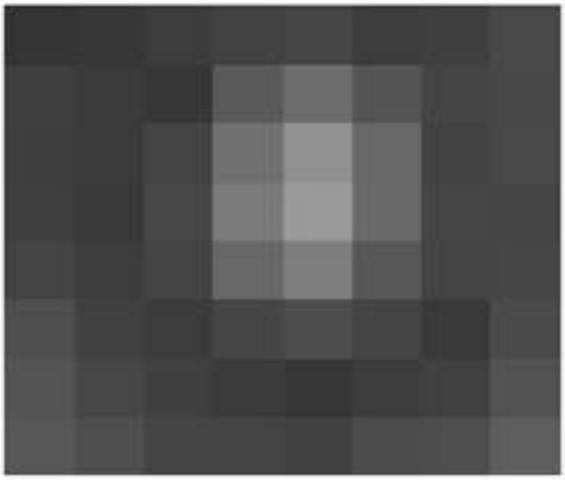


**Fig 8: Working of JPEG Compression**

Let's for the record, say that this 8x8 image contains the following values.

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

The range of the pixels intensities now are from 0 to 255. In order to change the range from -128 to 127, it is required to subtract 128 from each pixel value, we got the following results.

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Now we will compute using this formula.

$$G_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y} \cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & if\ n = 0 \\ \sqrt{\frac{2}{8}}, & Otherwise \end{cases}$$

The result comes from this is stored in let's say A(j,k) matrix.

This matrix is given below: -

$$\mathbf{Q_{j,k}} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Applying the following formula

$$B_{j,k} = round\left(\frac{A_{j,k}}{Q_{j,k}}\right)$$

We got this result after applying.

$$B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now ZIG-ZAG movement is performed on above matrix, whose sequence is shown below:

$$B_{j,k}=\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## 9.2 MPEG (MOVING PICTURE EXPERT GROUP)

MPEG is a method for video compression, which involves the compression of digital images and sound, as well as synchronization of the two. It also compress the sound track associated with the video. Algorithm used for MPEG compress the data into small bits for easy transmission and decompression and using "Discrete Cosine Transform" it can be encoded. MPEG simply store the change that has been made to the frames, so MPEG has high compression rate.

There currently are several MPEG standards: -

- MPEG-1 is designed for moderate data speed of up to 1.5 megabits per second.
- MPEG-2 is designed for high data speed of up to 10 Mbit/sec approx.
- MPEG-3 is designed for HDTV compression, but turned out to be redundant and integrated with MPEG2.
- MPEG-4 is designed for very low data rates less than 64 Kbit/sec.

i. A video is a temporal combination of frames, and a frame is a spatial combination of pixels..

ii. Compressing video, then, means spatially compressing each frame and temporally compressing a set off names.

iii. Spatial Compression: The spatial compression of each frame achieved with JPEG. Each frame can be independently compressed.

iv. Temporal Compression: In this type of compression, redundant frames are removed.

v. In temporally compress data, the first of all frames are divided into three categories by MPEG method

vi. I-frames, P-frames, and B-frames. Figure1 shows a sample sequence off names.

vii. Figure2 shows how I-, P-, and B-frames are constructed from a series of seven frames.

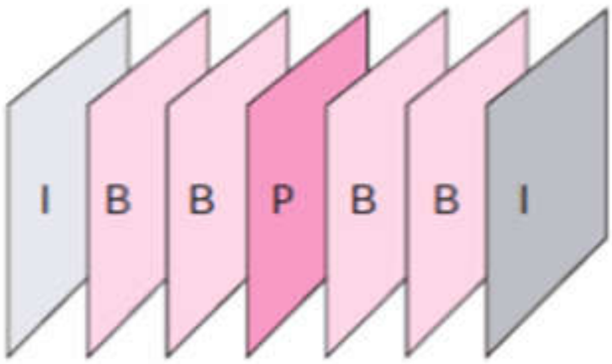



**Fig 9: MPEG frames**

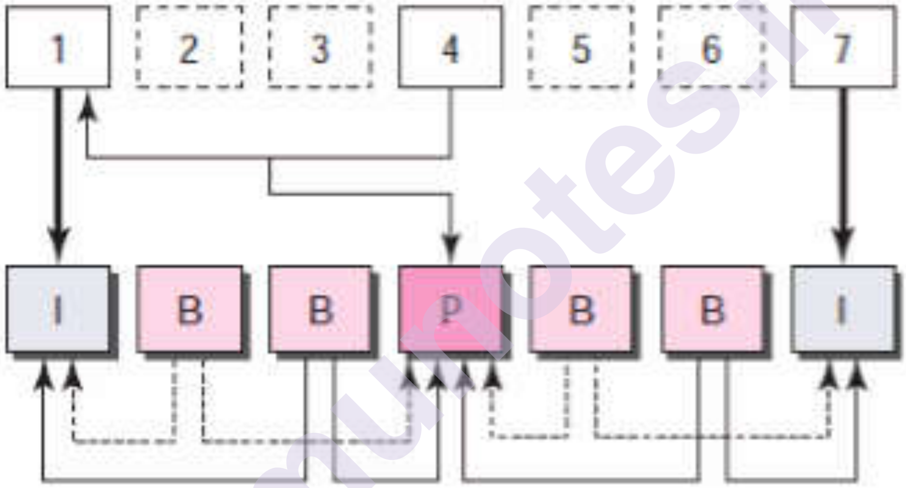



**Fig 10: MPEG frame construction**

**I-frames**: An inter frame (I-frame) is an independent frame i.e. different from other frame. This frame must appear handle some sudden change in the frame occasionally. A viewer can tune at any instance of time whenever a video is relayed. In case viewer tunes late, the viewer will not receive a complete picture at beginning of the broadcast.

**P-frames:** A predicted frame (P-frame) is related to the preceding I-frame or P-frame. In other words, each P-frame contains only the changes from the preceding frame. Previous I- or P-frames are only used to construct P-frames. As compared to other frame P-frame contains very much less information and even fewer bits after compression.

**B-frames:** A bidirectional frame (B-frame) is related to the preceding and following I-frame or P-frame. Note that a B-frame is different from another B-frame.

- ❖ The entire movie is designated a video sequence as per MPEG standard, and each picture has three components: one luminance component and two chrominance components (y, u & v).
- ❖ The luminance component contains the gray scale picture & the chrominance components provide the color, hue & saturation.
- ❖ The MPEG decoder has three parts, audio layer, video layer, system layer.
- ❖ The basic building block of an MPEG picture is the macro block as shown:

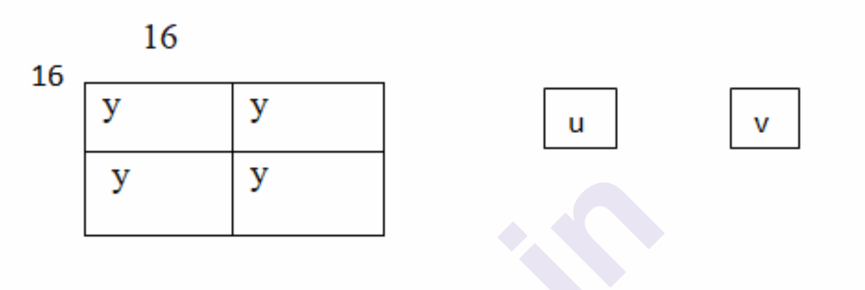



**Fig 11: Basic Building Block of an MPEG**

- ❖ The macro block consist of 16×16 block of luminance gray scale samples divided into four 8×8 blocks of chrominance samples.
- ❖ A macro block's MPEG compression consists of passing each of the °6 blocks through a JPEG-like DCT quantization and entropy encoding process.
- ❖ The MPEG standard defines a quantization stage having values (1, 31). Quantization for intra coding is:

$$Q_{\mathrm{DCT}} = \frac{(16 \times \mathrm{DCT}) + \mathrm{sign}(\mathrm{DCT}) \times \text{quantizer scale})}{2 \times \text{quantizer - scale} \times \theta}$$

Where,

Q = Quantization

DCT = Discrete cosine transform

Quantization rule for encoding,

$$Q_{\mathrm{DCT}} = \frac{16 \times DCT}{2 \times \text{quantizer - scale} \times \theta}$$

- The quantized numbers Q_(DCT )are encoded using non adaptive Huffman method.

## 9.3 VECTOR QUANTIZATION

Vector quantization being a non-transformed compression technique, is a powerful and efficient tool for lossy image compression. The idea of Vector Quantization (VQ) is to identify the frequently occurring blocks in a image and to represent them as representative vector and the set of representative vectors is known as Code Book and it is then used for image.

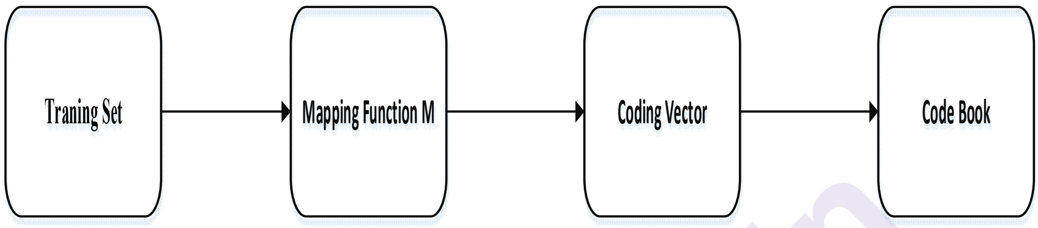



**Fig 12: Vector Optimization**

The goal of quantization usually is to produce a more compact representation of the data while maintaining its usefulness for a certain purpose. For example, to store color intensities you can quantize floating-point values in the range [0.0, 1.0] to integer values in the range 0-255, representing them with 8 bits, which is considered a sufficient resolution for many applications dealing with color. In this example, the spacing of possible values is the same over the entire discrete set, so we speak of uniform quantization; often, a non-uniform spacing is more appropriate when better resolution is needed over some parts of the range of values. Floating-point number representation is an example of non-uniform quantization—you have the as many possible FP values between 0.1 and 1 as you have between 10 and 100.

Both these are examples of scalar quantization—the input and output values are scalars, or single numbers. You can do vector quantization (VQ) too, replacing vectors from a continuous (or dense discrete) input set with vectors from a much sparser set (note that here by vector we mean an ordered set of N numbers, not just the special case of points in 3D space). For example, if we have the colors of the pixels in an image represented by triples of red, green, and blue intensities in the [0.0, 1.0] range, we could quantize them uniformly by quantizing each of the three intensities to an 8-bit number; this leads us to the traditional 24-bit representation.

By quantizing each component of the vector for itself, we gain nothing over standard scalar quantization; however, if we quantize the entire vectors, replacing them with vectors from a carefully chosen sparse non-uniform set and storing just indices into that set, we can get a much more compact representation of the image. This is nothing but the familiar paletted image representation. In VQ literature the "palette," or the set of possible quantized values for the vectors is called a "codebook," because you need it to "decode" the indices into actual vector values.

Figure 13 shows the result of this procedure applied to a grayscale version of the famous "Lena", a traditional benchmark for image-compression algorithms.

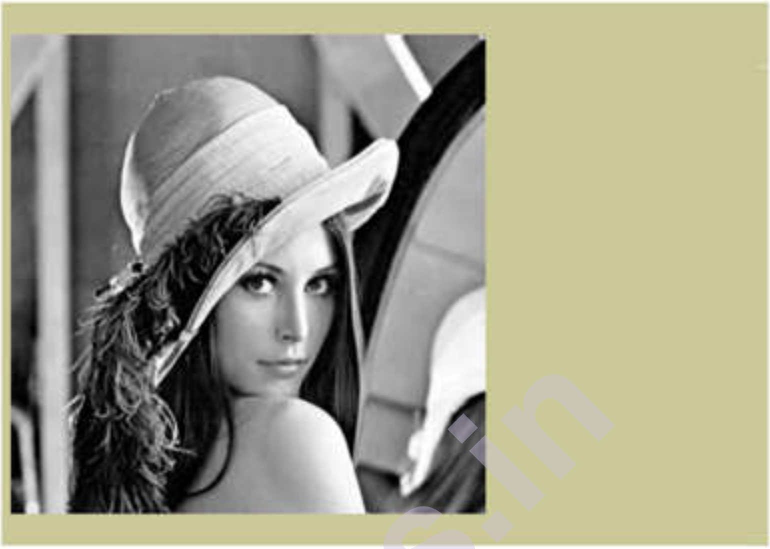


**Fig 13: Grey Scale Version**

The diagonal line along which the density of the input vectors is concentrated is the x = y line; the reason for this clustering is that "Lena," like most photographic images, consists predominantly of smooth gradients. Adjacent pixels from a smooth gradient have similar values, and the corresponding dot on the diagram is close to the x = y line. The areas on the diagram which would represent abrupt intensity changes from one pixel to the next are sparsely populated.

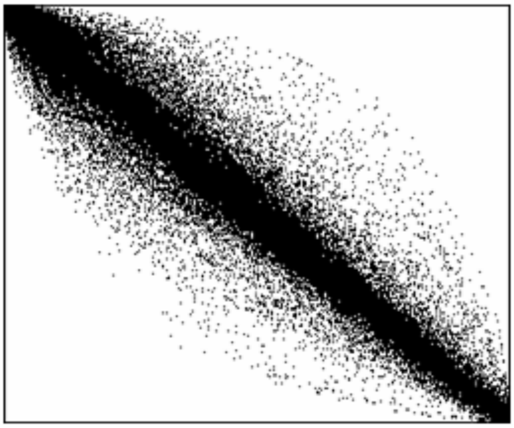


**Fig 14: Distribution of pairs of adjacent pixels from gray scale**

If we decide to reduce this image to 2 bits/pixel via scalar quantization, this would mean reducing the pixels to four possible values. If we interpret this as VQ on the 2D vector distribution diagram, we get a picture like Figure 15.

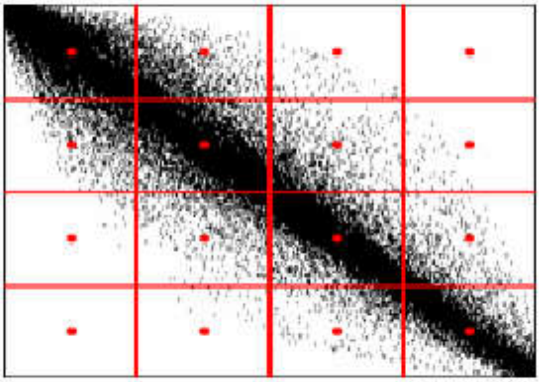


**Fig 15: Scalar quantization to 2 bits/pixel interpreted as 2D VQ.**

The big red dots on the figure represent the 16 evenly spaced possible values of pairs of pixels. Every pair from the input image would be mapped to one of these dots during the quantization. The red lines delimit the "zones of influence," or cells of the vectors—all vectors inside a cell would get quantized to the same codebook vector.

Now we see why this quantization is very inefficient: Two of the cells are completely empty and four other cells are very sparsely populated. The codebook vectors in the six cells adjacent to the x = y diagonal are shifted away from the density maxima in their cells, which means that the average quantization error in these cells will be unnecessarily high. In other words, six of the 16 possible pairs of pixel values are wasted, six more are not used efficiently and only four are O.K.

Let's perform an equivalent (in terms of size of resulting quantized image) vector quantization. Instead of 2 bits/pixel, we'll allocate 4 bits per 2D vector, but now we can take the freedom to place the 16 vectors of the codebook anywhere in the diagram. To minimize the mean quantization error, we'll place all of these vectors inside the dense cloud around the x = y diagonal.

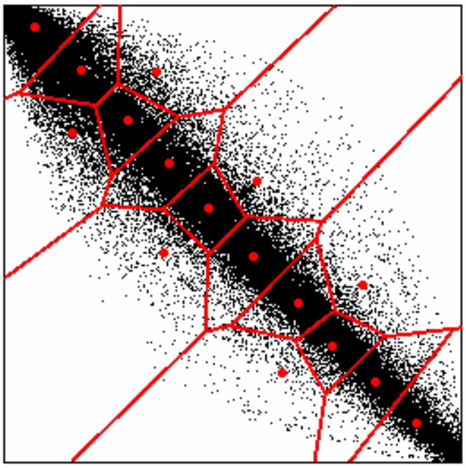


**Fig 16: Vector quantization to 4 bits per 2D-vector**

Figure 16 shows how things look with VQ. As in Figure 3, the codebook vectors are represented as big red dots, and the red lines delimit their zones of influence. (This partitioning of a vector space into cells around a predefined set of "special" vectors, such as for all vectors inside a cell the same "special" vector is closest to them, is called a Voronoi diagram; the cells are called Voronoi cells. You can find a lot of resources on Voronoi diagrams on the Internet, since they have some interesting properties besides being a good illustration of the merits of VQ.)

You can see that in the case of VQ the cells are smaller (that is, the quantization introduces smaller errors) where it matters the most—in the areas of the vector space where the input vectors are dense. No codebook vectors are wasted on unpopulated regions, and inside each cell the codebook vector is optimally spaced with regard to the local input vector density.

When you go to higher dimensions (for example, taking 4-tuples of pixels instead of pairs), VQ gets more and more efficient—up to a certain point. How to determine the optimal vector size for a given set of input data is a rather complicated question beyond the scope of this article; basically, to answer it, you need to study the autocorrelation properties of the data. It suffices to say that for images of the type and resolution commonly used in games, four is a good choice for the vector size. For other applications, such as voice compression, vectors of size 40-50 are used.

## 9.4 WAVELET BASED IMAGE COMPRESSION

Wavelet compression is a form of data compression well suited for image compression (sometimes also video compression and audio compression). Notable implementations are JPEG 2000, DjVu and ECW for still images, CineForm, and the BBC's Dirac. The goal is to store image data in as little space as possible in a file. Wavelet compression can be either lossless or lossy. Using a wavelet transform, the wavelet compression methods are adequate for representing transients, such as percussion sounds in audio, or high-frequency components in two-dimensional images, for example an image of stars on a night sky. This means that the transient elements of a data signal can be represented by a smaller amount of information than would be the case if some other transform, such as the more widespread discrete cosine transform, had been used.

Figure 17 shows a typical wavelet coding system. The various parameter such as an analyzing wavelet, c, and minimum decomposition level, $J - P$, are selected in order to encode a $2J \times 2J$ image. It is suitable to use fast wavelet transform, if the wavelet has a complementary scaling function w. In any of the case, a large portion of the original image is converted to vertical, horizontal, and diagonal decomposition by this transform. Many of the calculated coefficients contain very little visual information and can be quantized and coded to minimize redundancy. Moreover, to exploit any positional correlation across the $P$ decomposition levels the quantization can be adapted.

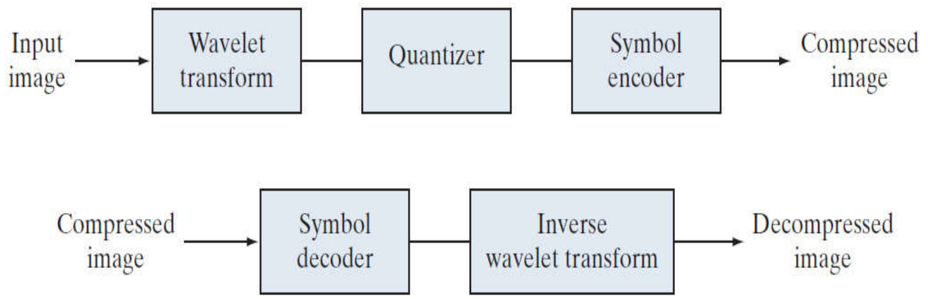



**Fig 17: shows a typical wavelet coding system**

## 9.5 MORPHOLOGICAL OPERATION

**Morphological image processing** is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to grey scale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a **structuring element**. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood:

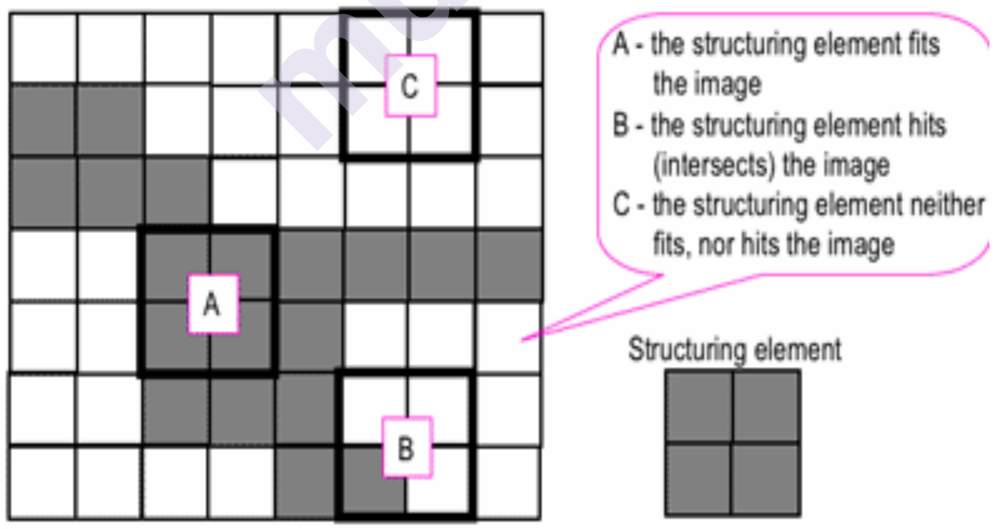



**Fig 18: Structuring Element**

**Dilation: -**

Dilation expands the image pixels for given element A by applying structuring element B. The equation of this operator is defined as

$$A \oplus B = \{(c+d) \mid for\ every\ c \in A, d \in B\}$$

A= Object to be dilated.

B=Structuring element.

**Steps to perform**

a) Fully match = 1

b) Some match = 1

c) No match = 0

**Example**

**Given image A**

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

**Structuring element B**

| |
|---|
| 1 |
| 1 |
| 1 |

**Output**

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |

**Erosion**

Erosion shrinks the image pixels for shrinking an element A by applying structuring element B. The equation of this operator is defined as: -

$$With\ A\ and\ B\ as\ set\ in\ z^2, the\ erosion\ of\ A\ by\ B\ defined\ as: -$$
$$A \ominus B = \{z | (B)_z \subseteq A\}$$

A= Object to be Eroded. B=Structuring element.

**Steps to perform**

a) Fully match = 1

b) Some match = 0

c) No match = 0

**For Example**

**Given image A**

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**Structuring element B**

| |
|---|
| 1 |
| 1 |
| 1 |

**Output**

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |

**Opening**

Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.

The *opening* of set *A* by structuring element *B*, denoted by $A \circ B$, is defined as: -

$$A \circ B = (A \ominus B) \oplus B$$

So, here **an erosion followed by a dilation.**

**For Example**

Set A

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Structuring Element B

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

**Output**

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**Closing**

Closing tends to smooth sections of contours but it generates fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

The *closing* of set *A* by structuring element *B*, denoted by $A^{\bullet} B$, is defined as: -

$$A^{\circ}B = (A \oplus B) \ominus B$$

So, here **dilation followed by a erosion.**

**For Example**

Set A

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Structuring Element B

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Output

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## 9.6 REFERENCES

1. Pratt WK. Introduction to digital image processing. CRC press; 2013 Sep 13.

2. Niblack W. An introduction to digital image processing. Strandberg Publishing Company; 1985 Oct 1.

3. Burger W, Burge MJ, Burge MJ, Burge MJ. Principles of digital image processing. London: Springer; 2009.

4. Jain AK. Fundamentals of digital image processing. Prentice-Hall, Inc.; 1989 Jan 1.

5. Dougherty ER. Digital image processing methods. CRC Press; 2020 Aug 26.

6. Gonzalez RC. Digital image processing. Pearson education india; 2009.

7. Marchand-Maillet S, Sharaiha YM. Binary digital image processing: a discrete approach. Elsevier; 1999 Dec 1.

8. Andrews HC, Hunt BR. Digital image restoration.

9. Lagendijk RL, Biemond J. Basic methods for image restoration and identification. InThe essential guide to image processing 2009 Jan 1 (pp. 323-348). Academic Press.

10. Banham MR, Katsaggelos AK. Digital image restoration. IEEE signal processing magazine. 1997 Mar;14(2):24-41.

11. Hunt BR. Bauesian Methods in Nonkinear Digital Image Restoration. IEEE Transactions on Computers. 1977 Mar 1;26(3):219-29.

12. Figueiredo MA, Nowak RD. An EM algorithm for wavelet-based image restoration. IEEE Transactions on Image Processing. 2003 Aug 4;12(8):906-16.

13. Digital Image Processing – Tutorialspoint. https://www.tutorialspoint.com/dip/index.htm.

14. Types of Restoration Filters. https://www.geeksforgeeks.org/types-of-restoration-filters/.

## 9.7 MOOCS

1. Fundamentals of Digital Image and Video Processing. Coursera. https://www.coursera.org/lecture/digital/mpeg-4-qYxK2

2. Moving Pictures Expert Group (MPEG) Video. SCTE. https://www.scte.org/education/course-offerings/course-catalog/moving-pictures-expert-group-mpeg-video/

3. Huffman Coding. Coursera. https://www.coursera.org/lecture/digital/huffman-coding-0CZoy

4. Morphology. Udemy. https://www.udemy.com/course/morphology/

## 9.8 VIDEO LINKS

- Fundamentals of Digital Image and Video Processing with Aggelos Katsaggelos. https://www.youtube.com/watch?v=6dJ6pitbuXE&list=PL3vl1rb9fAcLA7F38Qd9cqTuBNA20HclY

- Huffman Coding (Easy Example) | Image Compression | Digital Image Processing. https://www.youtube.com/watch?v=acEaM2W-Mfw

- Arithmetic encoding Digital Image Processing. https://www.youtube.com/watch?v=-vvgd87antk

- Image Compression Models | Digital Image Processing. https://www.youtube.com/watch?v=K807Ezea_GY

- LZW Coding | Digital Image Processing. https://www.youtube.com/watch?v=2FjOJMelZe0.

- How Image Compression Works. https://www.youtube.com/watch?v=Ba89cI9eIg8

## 9.9 QUIZ



1. Compressed image can be recover back by
(A) Image contrast
(B) Image enhancement
(C) Image equalization
(D) Image decomposition
**Answer: D**

2. What is the meaning of information ?
(A) Data
(B) Raw data
(C) Meaningful data
(D) None of these
**Answer: C**

3. Sequence of digital video is
(A) Frames
(B) Pixels
(C) Coordinates
(D) Matrix
**Answer: A**

4. What would you use compression for
(A) Making an image file smaller
(B) Modifying an image
(C) Both
(D) None of the above
**Answer: A**

5. Which of the following algorithms is the best approach for solving Huffman codes?
(A) Brute force algorithm
(B) Greedy algorithm
(C) Exhaustive search
(D) Divide and conquer algorithm
**Answer: B**

6. What is the running time of the Huffman encoding algorithm?
(A) O(log C)
(B) O(C)
(C) O(C log C)
(D) O(N log C)
**Answer: C**

7. Digitizing the image intensity amplitude is called
(A) Framing
(B) Sampling
(C) Quantization
(D) None of the above
**Answer: C**

8. Image compression comprised of
(A) Encoder
(B) Decoder
(C) Frames
(D) Both A and B
**Answer: D**

9. What is the full form of RLE ?
(A) Run line encoder
(B) Run length electrode
(C) Run length encoding
(D) None of the above
**Answer: C**

10. Which bitmap file format support the Run length encoding ?
(A) BMP
(B) PCX
(C) TIF
(D) All of the above
**Answer: D**

11. In Huffman coding, data in a tree always occur?
(A) Roots
(B) Leaves
(C) Left sub trees
(D) Right sub trees
**Answer: B**

12. Which of the following of a boundary is defined as the line perpendicular to the major axis?
(A) Minor axis
(B) Median axis
(C) Equidistant axis
(D) Equilateral axis
**Answer: C**

13. The order of shape number for a closed boundary is:
(A) Even
(B) Odd
(C) 1
(D) Any positive value
**Answer: A**

14. Which of the following techniques of boundary descriptions have the physical interpretation of boundary shape
(A) Laplace transform
(B) Fourier transform
(C) Statistical moments
(D) Curvature
**Answer: C**

15. What does the total number of pixels in the region defines?
(A) Area
(B) Intensity
(C) Brightness
(D) None of the above
**Answer: A**

16. For which of the following regions, compactness is minimal?
(A) Square
(B) Irregular
(C) Disk
(D) Rectangle
**Answer: C**

17. On which of the following operation of an image, the topology of the region changes?
(A) Rotation
(B) Folding
(C) Stretching
(D) Change in distance measure
**Answer: B**

18. Which of the following techniques is based on the Fourier transform?
(A) Spectral
(B) Structural
(C) Topological
(D) Statistical
**Answer: A**

19. Based on the 4-directional code, the first difference of smallest magnitude is called as:
(A) Chain number
(B) Difference
(C) Difference number
(D) Shape number
**Answer: D**

20. What is the unit of compactness of a region?:
(A) Meter
(B) Meter2
(C) Meter-1
(D) No units
**Answer: D**

❖❖❖❖

# Module VI

# 10

# APPLICATIONS OF IMAGE PROCESSING

**Unit Structure**



## 10.1 CASE STUDY ON DIGITAL WATERMARKING

Digital watermarking is a technology that embeds data into digital multimedia content, verifying content reliability and is used to recognize owner ID [1].

Digital watermarks hide copyright information in digital data through specific algorithms. The secrecy information embedded is some text, author number, company logo, especially important photos. This secret information is embedded in digital data (image, audio, and video) to ensure security, data authentication, owner identification, and copyright protection. The watermark is either display or visible to digital data. You need to apply good water sharing technology to strongly embed the watermark. Fig 1. shows Digital Watermark embedding process and Fig. 2. shows watermark detection process.

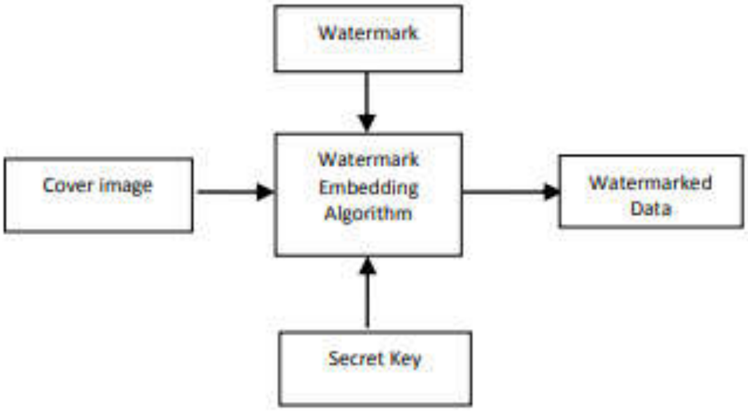



**Fig 1: Watermarking embedding process [2]**

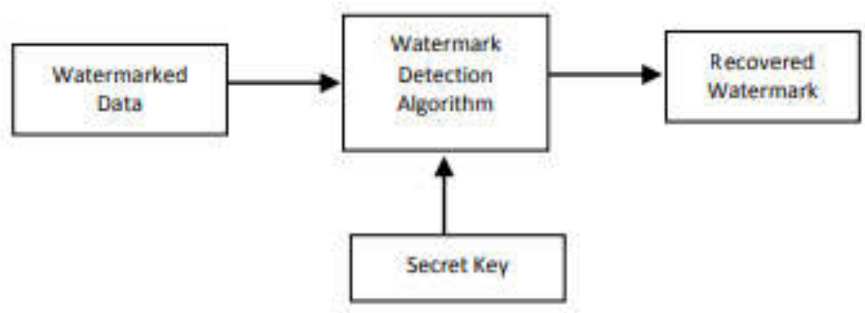



**Fig 2: Watermark Detection Process [2]**

**Digital watermarking process (Life cycle) [3]:**
The process consists of 3 main parts:

1. **Embed**
2. **Attack**
3. **Protection**

**Embed**: Embedded with the digital watermark.

**Attack**: Any change in the transmitted content, it becomes a threat and is called an attack to the watermarking system.

**Protection**: The detection of the watermark from the noisy signal which might have altered media is called Protection.

Types of Watermarks [3]:

1. Visible Watermarks
2. Invisible Watermarks
3. Public Watermarks
4. Fragile Watermarks

**Visible Watermarks:** These are visible in nature.

**Invisible Watermarks:** These are invisible but are embedded in the media and use steganography technique.

**Public Watermarks:** These can be modified using certain algorithms by anyone and are not secure.

**Fragile Watermarks:** These are said to be destroyed as data manipulation occurs, need to use a system as to detect the changes occurred to the data, if fragile watermarks are used.

Digital watermarking is used for numerous purposes including [1-2]:

- ❖ Broadcast Monitoring
- ❖ Ownership Assertion
- ❖ Transaction Tracking
- ❖ Content Authentication
- ❖ Copy Control and Fingerprinting

Types of digital watermarking [1]:

- ❖ Visible Digital Watermarking
- ❖ Invisible Digital Watermarking

**Visible Digital Watermarking:** It is embedded as the watermark and can be used as a logo or as a text representing the owner [1].

**Invisible Digital Watermarking:** The data embedded is invisible.

Example 1: Audio is inaudible in case of an invisible audio content.

Example 2: Image/text is not visible in the case of an invisible text/image/ multimedia content.


Fig 3: (a) Original fingerprint image (b) Watermarked fingerprint image

## 10. 2 DIGITAL WATERMARKING TECHNIQUES: A CASE STUDY IN FINGERPRINTS AND FACES

The purpose of watermarks is two-fold:

(i) Used to determine ownership, and

(ii) Used to detect tampering.

There are essential characteristics that a watermark must have and must be detectable. To determine ownership, it is required to be able to retrieve the watermark. There are basically two mechanisms by which a watermark can be retrieved. The incomplete watermark can only be restored if the original image is present. The full watermark can be retrieved independently. Full watermarks are more desirable as they apply to more applications. When watermarking large files or a large number of files in a database, full watermarks are preferred because they avoid storing multiple copies of the original file. Second, the watermark should be robust to many different types of signal processing. If the watermark is not strong, it will be useless as the assets will be lost during processing. Having some built-in fragile features can sometimes be helpful. If fragile watermarks are used and the data is altered, the watermark can identify areas that have been altered. Fragile watermarks can detect minor changes or tampering of data. On the other hand, strong watermarks are useful for detecting large-scale attacks on data.

Various watermark schemes have been developed. One of the first watermarking algorithms was to manipulate the least significant bit (LSB) of pixels in the spatial domain [6]. There are many ways to apply LSB schemes where all LSBs can be changed or a random set of LSBs can be changed. These diets are especially helpful because of their fragility. If a person modifies an image, it is more likely that the LSB will also be modified. Unfortunately, it is this fragility that can cause a host of other problems. It deletes all watermarks. If enough LSB is changed, the watermark will be unrecoverable. Furthermore, it is possible to modify the image without changing the LSB. If this is done, the watermark is essentially useless, as it cannot be used for tampering detection.

In general, spatial (pixel) domain schemas are too fragile to withstand an attack. Resulting in the development of solutions in the frequency domain. There are two general algorithms in the frequency domain, a spreading method and a block method. Basically, the DCT of the entire image is captured and a watermark is applied to the preselected frequencies. If the DCT image is represented by $V(j,k)$ and the watermark is $W(j,k)$, then the watermark is $V^*(j,k) = V(j,k) + \alpha W(j,k)$ , where $W(j, k)$ is normally distributed and $\alpha$ is a scale parameter. In the simplified version of the method, the value of alpha is fixed at 0.1. For better results, $\alpha$ can be inferred from the JND (just a notable difference) matrix. The JND matrix containing the mean can be added to each pixel without causing noticeable perceptual changes in image.

DCT block method is another method used. It is similar to the spectral spectrum method, but instead of taking the DCT of the entire image, the DCT is taken for 8x8 blocks (or 16x16). This method allows the position of the fuzzy. It also has the advantage because it is compatible with relevant compression techniques such as MPEG. It can be built directly in MPEG processor. However, it has its drawbacks because it is applied separately for each image segment; It's easier to remove this type of watermark.

With the popularity of JPEG format, the development of strong compression watermarks is a major concern. There are some powerful filigree algorithms with compression. In these modes, watermarks are often inserted into the frequency field of compressed images. When the watermark is placed in uncompressed images, the watermark can be broken during the compression process. In fact, it can be so damaged that it's unrecognizable. But by inserting the watermark into the compressed frequency domain, the compression will have little effect on the watermark. By placing multiple watermarks in the same image, the ability to determine if the image has been tampered with and where the image has been tampered with increases. Usually, two watermarks are placed in one image. One of the watermarks has powerful image processing capabilities, and the other watermark accurately detects small changes in the image (i.e. fragility). In previous studies, a watermark was inserted into the frequency domain (watermark) as well as in the Pixel field (Watermark fragile). There are a number of disadvantages for this type of diagram, because two odds are inserted, there is no maximum value possible at its maximum value. Required is their intensity at scale. Also, when inserting watermarks, it is essential to first insert Filigree powerful. If the fragile fuzz is inserted first, then as soon as the watermark is definitely inserted, the fragile watermark will detect Alleviant change!

**Proposed Technique**

A number of different watermarking schemes are in use today, for each there is a simple mechanism to detect fraud and determine ownership. There are times when an image owner wants to pass an image on to someone else. In this case, how can the recipient ensure that the received image is not corrupted? It is clear that some keys must be used. At the most basic level, you can use a watermark or an original image as a key. But either way would be a bad choice. A watermark is similar to a PIN or password. Granting access to others can be detrimental as they can take ownership of your images and remove them. Sending the original image itself defeats the purpose of the watermark. This is because the recipient can now transfer ownership by watermarking another watermark on the original image. Ideally, the key used should be unique for each image. This will prevent smart hackers from attacking you. An idea that comes to mind is block characterization of the image. One potential key you can use is the regional mean matrix. The local average of the transferred images is computed as a 5x5 block and can be used as a key. The receiver can compare the key to a set of local averages computed over the received images. This makes it easy to verify the authenticity of an image. This key is attractive in that it is a fraction of the size of the original image, allowing you to pinpoint where the change occurred in pixels.

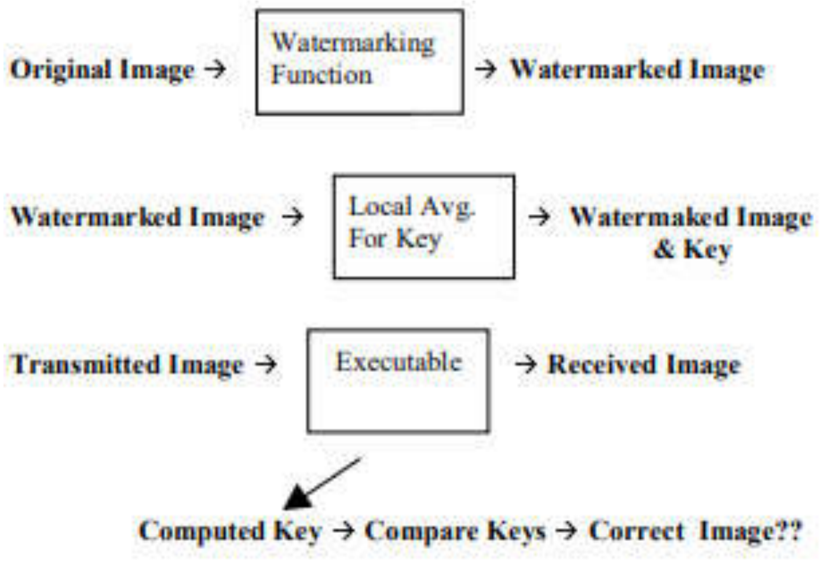



**Fig 4: Watermarking Technique**

The local average technique was used to detect image tampering in different scenarios:

(i) smudging,

(ii) compression, and

(iii) Wiener filtering for a number of images.

An executable was passed to allow the receiver to generate a key for the image and compare it with the real key. The keys are essentially an exaggerated version of the local 5x5 average block. Magnification makes it easy to detect small differences, allowing activities such as compression to be detected. A threshold must be established that must be a function of the key's magnification. Smudging is a type of image damage that has been investigated. It is easily detected in fingerprints and faces using a local mean-based key. Looking at the actual key and the key generated, it's almost clear where the change happened. But if he is not convinced, the user can compare the keys numerically and locate the modified image and how much.

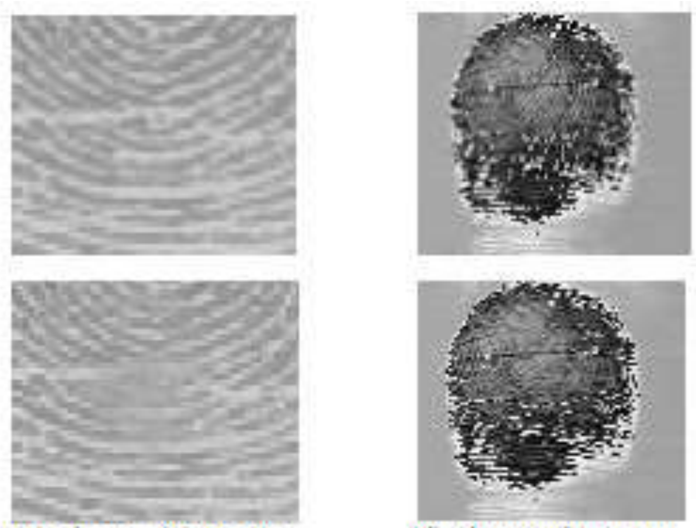



**Fig 5: Detecting smudging in a fingerprint image. Figure (c) shows a magnified version of the area that was smudged. The original can be seen in figure (a). Figures (b) and (d) show the keys for figures (a) and (b), respectively**



**Conclusion**

Watermarking biometric information may be a still a comparatively new issue, however it's of growing importance as a lot of sturdy strategies of verification and authentication are being used. Biometrics give the mandatory distinctive characteristics but their validity should be ensured. A receiver can't perpetually confirm whether or not or not she has received the right data while not the sender giving her access to important data like watermark. The key projected here is one amongst several potential methods. The native average theme creates a semi-unique key for every data set transmitted and so is tougher to tamper with. It conjointly has the flexibility to pin-point wherever meddling has occurred up to a small pixel window and information security will be assured in databases similarly as in transmission. However, it's solely a semi-unique key. It’s do able to change the image however retain constant key, because the average isn't perpetually the simplest tool for characterizing data. A non-linear mechanism may well be more sensitive to small changes and are a few things that might be investigated. One major flaw in our methodology is its inability to notice whether or not the alterations within the image are because of channel distortions and noise or actual tampering by an individual. Generally the transmission noise may be a perform of the encryption theme employed, and at alternative times it's a function of the channel itself. However, having how to work out whether or not the “tampering” is that the results of noise or a malicious attack would be useful. As for the noise to be seen as tampering, it should be sturdy enough to start out disrupting the image and from that time on that may be understood as an accidental attack. Another potential drawback is the “disgruntled worker” attack. If a discontented employee has access to the executable, then it is straightforward to form the possible perpetually agree that the image received has not been tampered albeit it's been.

Similarly, the executable may be altered in order that it provides systematically negative responses. a technique to try and do thus would be to introduce a random perform that operates in conjunction with the executable, so that for example, a 5x5 native average key's not the sole possibility.

## 10.3 VEHICLE REGISTRATION NUMBER PLATE DETECTION AND RECOGNITION USING IMAGE PROCESSING TECHNIQUES [6]

The objective of the proposed work is the application of new techniques of image segmentation and other processing techniques in the context of the identification and production of license plates. The prerequisite is that the plates are in the following format: TS 16 EX 5679, where the first two characters indicate the registration State of the vehicle. First, the license plate region (the region of interest) must be located and extracted from the larger image of the acquired vehicle.

In this work, different image processing techniques are used in the pre-processing phase, namely morphological transformation, Gaussian smoothing, Gaussian threshold. Then, for plate segmentation, the outlines are applied following the edge and the outlines are filtered based on font size and location space. Finally, after the region of interest filtering and straightening area, the Knearest neighborhood algorithm is used for character recognition.The main contributions of this work are the design of an Indian vehicle license plate detection and recognition system using an image processing system to address the following challenges:Dealing with varying illuminated images.

- ❖ Dealing with bright and dark objects
- ❖ Dealing with noisy images.
- ❖ Dealing with non-standard number plates
- ❖ Dealing with cross-angled or skewed number plates
- ❖ Dealing with partially worn our number plates.

**Methodology**

The proposed methodology consisting of three major phase’s viz., pre-processing, detection, and recognition are shown in Figure 1.

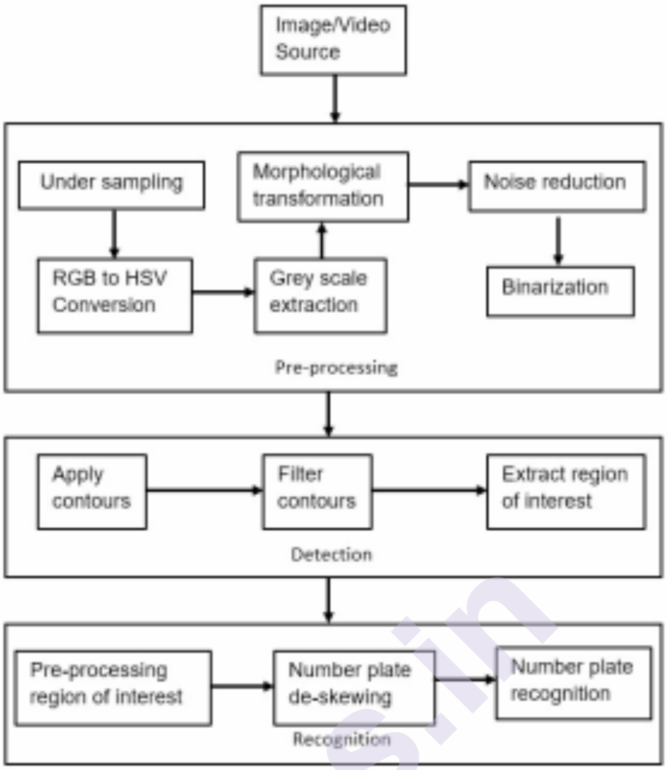



**Fig 6: The proposed number plate recognition system**

## PRE-PROCESSING

The input can  be an image or a video. Video is considered as a series of frames/frames, before starting license plate detection, the image source must be matched for further processing. Figure 7(a) is the example input image used to show the process. Here is the order in which the image processing techniques are applied:Image Under-Sampling

- ❖ RGB to HSV Conversion
- ❖ Grayscale extraction
- ❖ Morphological transformations
- ❖ Gaussian Smoothing
- ❖ Inverted Adaptive Gaussian Thresholding

At the end of the previous stage of image pre-processing, Inverted Adaptive Gaussian Thresholding, returns a binarized image, with values of either 0 or 255.

### Training the model

The K Nearest Neighbors (KNN) algorithm was used to train the model. Many other models like Decision Tree, Gradient Boosting have been tested, but K Neighbors got better results. To extract the best possible hyper-parameters for the model, a random search was used. Randomized search is an optimized version of parameter sweep or grid search, in which a rigorous search is performed from a manually formed space of a subset of hyper-parameters belonging to the learning algorithm. Performance metrics are used in the guidance of grid search such as, cross-validation of the training-set or evaluation of the validation-set. The parameter space explored by grid search and random search is the same. Setting the parameters is quite similar; however the execution time in case of randomized search is much shorter.

Before saving the given character/font, it is transformed to a standard size of 20 x 30 pixels. This ensures the consistency of model inputs. Figure 7(a) shows the characters used and Figure 7(b) depicts the extracted images for the specified character 'P'.

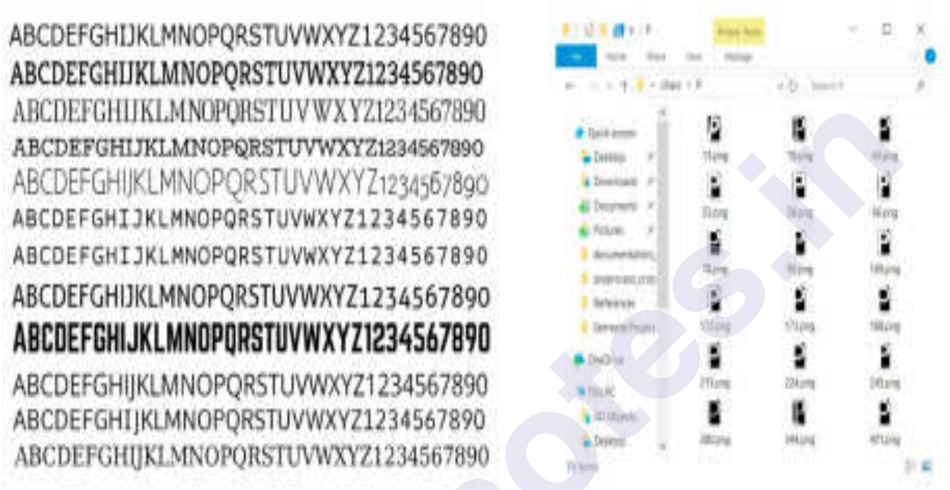



**Fig. 7. (a) Fonts used for training; (b) Extracted images for a the letter 'P'**

**Results and discussion**

The experiments **were** conducted on a Windows 10 machine with **8 GB of RAM** and an i5 processor **running at** 2.4 GHz frequency. The OpenCV Python library is used to implement image processing tools. System testing was performed with photos and videos. All of the above cases, such as irregularly illuminated plates, stylized fonts, close-up plates, far away plates are considered part of the testing including images with different environmental conditions. Figure 8 (a) shows an image for testing the case of irregular and small number plates. Figure 8 (b) shows a case of partially worn-out and a standard number plate.


**Fig. 8. (a) Irregular illumination and small number plate; (b) A partially worn out number plate; (c) standard number plate**

**CONCLUSION**

The job involves detecting number plates and recognizing the number plate, involving the number of Indian vehicles or number plates. The main contributions of this work include: taking into account difficult situations such as light changes, blur, asymmetrical, noisy, no standard images and partial worn plates. In this job, first, some image processing techniques, morphological changes, Gaussian smoothing, the Gaussian threshold is used in the pre-handling period. Then, for the segment of the number plate, the borders are applied at the next boundary and the contours are filtered according to the size of the character and space location. Finally, after filtering and transforming areas of interest, neighborhood algorithms are used to recognize the characters.

## 10.4 OBJECT DETECTION USING CORRELATION PRINCIPLE [7]

The problem definition of object detection is to determine where objects are located in a given image and which category each object belongs to. So the pipeline of traditional object detection models can be mainly divided into three stages:

❖ Informative region selection,

❖ Feature extraction and

❖ Classification.

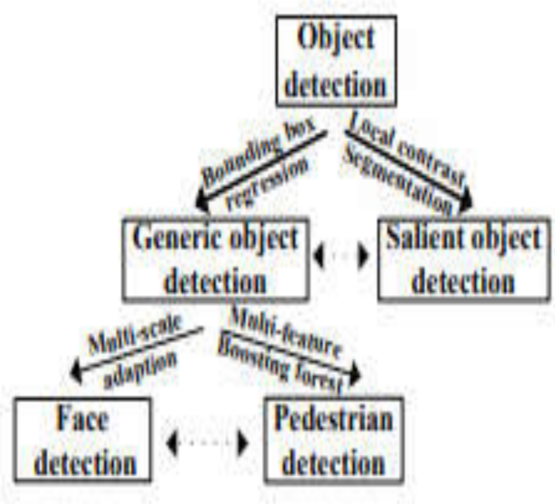



**Fig 9: The application domains of object detection [7].**

Deep Neural Networks (DNNs), more profitable to introduce regions with CNN (RCNN) functions. DNNs, or most typical CNNs, operate quite differently from traditional approaches. They have deeper architectures with the ability to learn more complex features than shallower ones. In addition, expressiveness and powerful training algorithms make it possible to learn information object representations without manually designing features.

Deep learning has been popular since 2006 with a breakthrough in speech recognition. The recovery of deep learning can be attributed to the following factors.

1. The emergence of large-scale annotated training data, such as ImageNet, to fully demonstrate its enormous learning capacity;

2. The rapid development of high-performance parallel computing systems, such as GPU clusters;

3. ignificant advances in the design of network structures and training strategies. With unsupervised and layer wise pre-training guided by Auto-Encoder (AE) or Restricted Boltzmann Machine (RBM), a good initialization is provided.

CNN advantages against traditional methods are summarized as follows.

Hierarchical feature representation, which is the multilevel representations from pixel to high-level semantic features learned by a hierarchical multi-stage structure, can be learned from data automatically and hidden factors of input data can be disentangled through multi-level nonlinear mappings.

Compared with traditional shallow models, a deeper architecture provides an exponentially increased expressive capability.

The CNN architecture provides an opportunity to optimize numerous related tasks jointly.

Benefitting from the large learning capacity of deep CNNs, some classical computer vision challenges can be recast as high-dimensional data transform problems and solved from a different viewpoint.

Due to these advantages, CNN has been widely applied into many research fields, such as image super-resolution, reconstruction, image classification, image retrieval, face recognition, pedestrian detection and video analysis.

**Region Proposal Based Framework**

Region-proposal-based framework, a two-step process, corresponding to some degree for the human brain's attention mechanism, which first provides a rough analysis set about the entire scenario, then focus on regions of interest. Among the works related to the earlier, the most typical is Overfeat. This model inserts CNN into the sliding window method, which predicts the bounding boxes directly from the top positions of the feature map after obtaining the confidants of confidences of underlying object categories.

RCNN: It is important to improve the quality of candidate bounding boxes and apply a deep architecture to extract high-level functionality. To address these issues, RCNN was proposed by Ross Girshick in 2014 and achieved a mean average accuracy (mAP) of 53.3% with an improvement of more than 30% over the previous best. (DPM HSC) on PASCAL VOC 2012. Figure 10 shows the flowchart of the RCNN, which can be divided into three phases as follows.

**Region proposal generation**

The RCNN uses a selective search to generate approximately 2,000 region proposals for each image. The selective search method relies on simple bottom-up clustering and salience indices to quickly provide more accurate candidate boxes of arbitrary size and to reduce the search space in object detection. feature extraction. At this point, each proposed region is warped or cropped to a fixed resolution and the CNN module is used to extract a 4096 dimensional feature as the final representation. Due to the high learning capacity, dominant expressive power and hierarchical structure of CNNs, it is possible to obtain a high-level, semantic and robust representation of the features of each proposed region.

**Classification and localization**

With pre-trained linear category-specific SVMs for multiple classes, the different region proposals are evaluated against a set of positive regions and background (negative) regions.

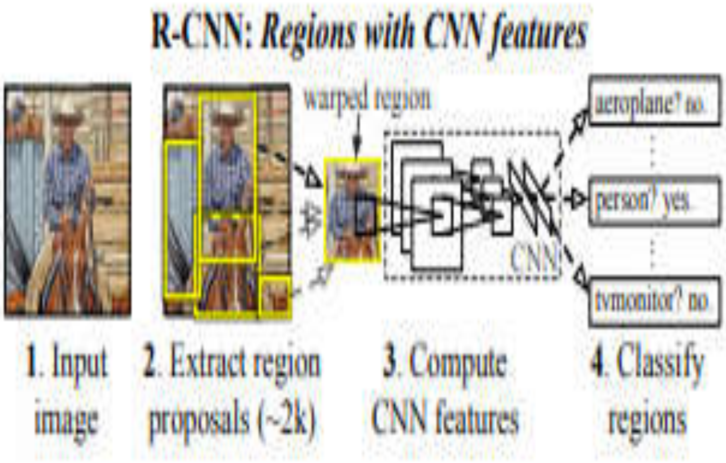



**Fig 10: R-CNN – Regions with CNN features**

Face detection is essential for many facial applications and serves as an important pre-processing procedure for face recognition, face synthesis and facial expression analysis.Unlike general object detection, this task involves recognizing and locating regions of the face that cover a very wide range of scales (30,300 points versus 101,000 points). ; impose great challenges in real applications. The most famous facial detector proposed by Viola and Jones trains cascade classifiers with HaarLike and AdaBoost features, achieving good performance with real-time efficiency. In contrast to this cascading structure, Felzenszwalb et al. proposed a deformable part model (DPM) for face detection. However, for these traditional face detection methods, high computational costs and large amounts of annotations are required to achieve a reasonable result. Moreover, their performance is severely limited by hand-designed features and surface architecture.

Despite rapid development and promising advances in object detection, there are still many open questions for future work. The first is the detection of small objects as done in the COCO dataset in the face detection activity. To improve the localization accuracy on small objects in case of partial occlusions, it is necessary to modify the network architectures of the following aspects.

❖ Multi-task joint optimization and multi-modal information fusion.

❖ Scale adaption

❖ Spatial correlations and contextual modeling

The second one is to release the burden on manual labor and accomplish real-time object detection, with the emergence of large-scale image and video data. The following three aspects can be taken into account.

**Unsupervised and weakly supervised learning**

The third one is to extend typical methods for 2D object detection to adapt 3D object detection and video object detection, with the requirements from autonomous driving, intelligent transportation and intelligent surveillance.

- 3D object detection
- Video object detection

**CONCLUSION**

Due to its powerful learning ability and advantages in dealing with occlusion, scale transformation and background switches, deep learning based object detection has been a research hotspot in recent years. Review starts with generic object detection pipelines which provide base architectures for other related tasks. Then, three other common tasks, namely salient object detection, face detection and pedestrian detection, are also briefly reviewed. Finally, several promising future directions to gain a thorough understanding of the object detection landscape have been expressed. This section is meaningful for the developments in neural networks and related learning systems, providing valuable insights and guidelines for future progress.

❖❖❖❖

# 11

# HUMAN BODY TRACKING BASED ON DISCRETE WAVELET TRANSFORM

**Unit Structure**



## 11.1 HUMAN BODY TRACKING BASED ON DISCRETE WAVELET TRANSFORM [8]

A novel human body tracking system based on discrete wavelet transform is proposed based on color and spatial information. The configuration of the proposed tracking system is very simple, consisting of a CCD camera mounted on a rotary platform for tracking moving objects. By using the position information of objects in the image frame captured by the camera, the rotary platform is controlled to position the tracking object around the central area of images to improve tracking efficiency.

Image tracking is getting more and more popular over the past years because of the advancements of automation technologies and is widely applied in surveillance systems, robot localization, human computer interaction, etc. Researches of human-body tracking are the most attractive one although difficulties still exist because the shapes and dynamics of humans are complicated and the backgrounds are cluttered. Over the past years, many applications of people tracking systems such as surveillance, human-computer interface, people counting system, etc, have been attempted based on a popular method of background subtraction to segment and track moving objects in real-time surveillance. Segmentation methods using background subtraction, however, have difficulties in image sequences from moving camera or sequences including instantaneous change of illumination or shadow.

## Discrete wavelet transform

Discrete Wavelet transform has been extensively applied in the areas of image processing, image compression, edge detection, and texture analysis. A 2-dimensional wavelet transform decomposes an image into 4 sub-images as shown in Fig. 11, where filters are first applied in one dimension (e.g. X axis) and then in the other (e.g. Y axis). Because down sampling is performed at these two stages, the size of the sub-images becomes 1/4 as large as the original image. Observing these four sub-images in Fig. 12, we found that the wavelet transform preserves not only the frequency features but also spatial ones. It can be decomposed into four different bands (LL, HL, LH, HH) via the discrete wavelet transform. These sub-bands contain different frequency characteristics with the use of high-pass and low-pass filters. The high-pass filter extracts the high-frequency portions (e.g. edges of the object). On the other hand, the low-pass filter gives the low-frequency information representing the most energy of an image and rejects the noise of an image as well. The basic idea is to use the wavelet transform to reduce the resolution of each frame of the sequence for reducing the computational cost. Basis wavelet transform is used due to its simplicity and speed efficiency, where only the low-frequency part is used for processing due to the consideration of low computing cost and noise reduction issue. The original image of 240× 320 is pre-processed via a 2-level discrete wavelet transform to obtain a lowest-frequency sub-image (i.e. LL2 in Fig. 12(d)) for further processing in the proposed tracking system. As a result, the image size of the sub-image LL2 has been reduced to 60×80 , which represents 1/6 of the size of the original image.

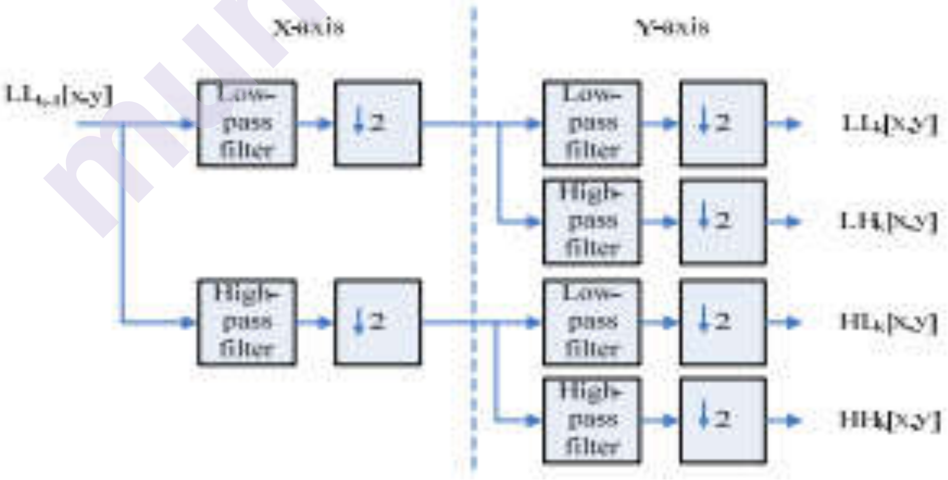



**Fig 11: Two-dimensional discrete wavelet transform**

## The proposed human body tracking system

The objective of tracking is to closely follow objects in each frame of a video stream such that the object position as well as other information is always known. To overcome difficulties in achieving realtime tracking and improving tracking efficiency, a novel colour-image real-time human body tracking system based on discrete wavelet transform, where a CCD camera is mounted on a rotary platform for tracking moving objects. Procedures in tracking moving objects via the proposed approach can be illustrated via a flowchart shown in Fig. 13.

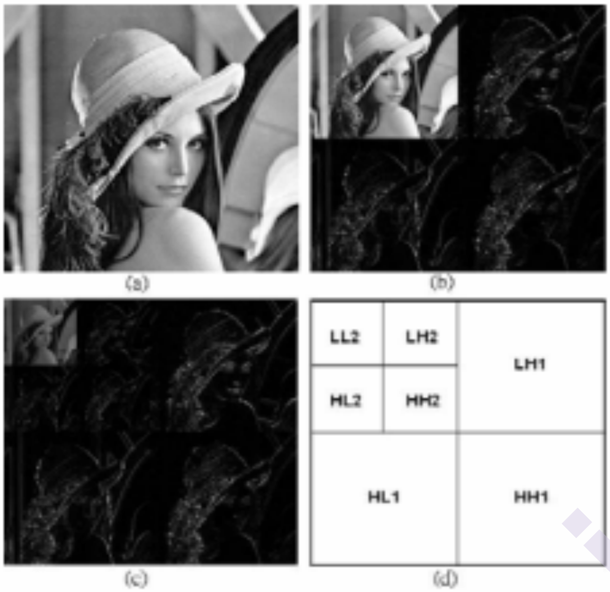



**Fig 12: (a) Original image (b) first-level DWT (c) second-level DWT (d) sub-bands of second-level DWT**

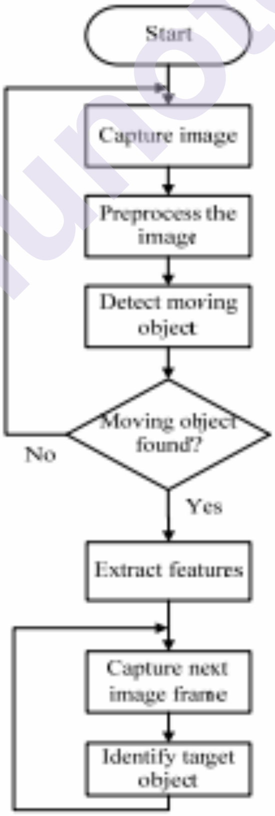



**Fig 13: Flowchart of the tracking procedures**

**Experimental results**

Proposed tracking system is implemented in Windows XP PC with Pentium 2.0G CPU, 1024MB RAM under Borland C++ builder 5 software environment as the implementation platform. The resolution of each color images is 320x240 pixels. We can achieve real-time processing at about 25 frames per second. As demonstrated satisfactory performances have been achieved via the proposed approach.

**Conclusions**

With the aim at single human-body tracking, a novel colour image real-time human body tracking system based on discrete wavelet transform is proposed for identifying the target based on color and spatial information. To improve tracking performances, discrete wavelet transform is used to pre-process the image for reducing computations required and achieving real-time tracking. The experiments results have shown that the proposed tracking system is capable of realtime tracking human objects in about 25 frames per second.

## 11.2 HANDWRITTEN AND PRINTED CHARACTER RECOGNITION

Indian script is collection of scripts used in the sub-continent namely Devanagari, Bangla, Hindi, Gurmukhi, Kannada and etc. The researchers used data that was already in an isolated form in order to avoid the segmentation phase and are based on statistical and structural algorithms. The results of Devanagari scripts were found to be better than English numerals. Devanagari had a recognition rate of 89% with 4.5 confusion rate, while English numerals had a recognition rate of 78% with confusion rate of 18%. A modular neural network was used for script identification while a two-stage feature extraction system was developed, first to dilate the document image and second to find average pixel distribution in the resulting images. The researchers used 64 directional features based on chain code histogram for feature recognition. The proposed scheme resulted in 98.86% and 80.36% accuracy in recognizing Devanagari characters and numeral, respectively. Five-fold cross-validation was used for the computation of results. Perwej and Chaturvedi used backpropagation based neural network for the recognition of handwritten characters and the results showed that the highest recognition rate of 98.5% was achieved. Obaidullah et al. proposed Handwritten Numeral Script Identification or HNSI framework based on four indices scripts, namely, Bangla, Devanagari, Roman and Urdu. The researchers used different classifiers, namely NBTree, PART, Random Forest, SMO, Simple Logistic and MLP and evaluated the performance against the true positive rate. Performance of MLP was found to be better than the rest. Research on Indian scripts is very diverse, and a number of researchers are involved in research on multiple scripts. This is the reason why a number of research articles on character recognition of Indian scripts are growing each year. researchers have used techniques like Tesseract OCR and google multilingual OCR, Convolutional Neural Network (CNN) Deep

Belief Network with the distributed average of gradients feature, Modified Neural Network with the aid of elephant herding optimization, VGG (Visual Geometry Group) and SVM classifier with the polynomial and linear kernel.

**CEDAR**

CEDAR, was developed by the researchers at the University of Buffalo in 2002 and is considered among the first few large databases of handwritten characters. In CEDAR, the images were scanned at 300 dpi as shown in Figure 14.

**Fig 14: CEDAR dataset**

**CHARS74K**

Chars74k dataset was introduced by researchers at the University of Surrey in 2009 which contains 74,000 images of English and Kannada (Indian) scripts. Segmentation of individual characters was done manually, and results were presented in bounding box segmentation. Bag of visual words technique was used for object categorization, and eventually, 62 different classes were created for English and 657 classes for Kannada. A number of researchers have used CHARS74k dataset for recognition of Kannada script. It is to be noted that Kannada is one of many Indian scripts we have included in this research. There are various datasets for Indian language, depending on the script that has been used.

Fig 15: Sample image from CHARS74K dataset

**CONCLUSION**

1) Optical character recognition has been around for the last eight (8) decades. Development of machine learning and deep learning has enabled individual researchers to develop algorithms and techniques, which can recognize handwritten manuscripts with greater accuracy.

2) Systematically extracted and analyzed research publications on six widely spoken languages. We explored that some techniques perform better on one script than on another, e.g. multilayer perceptron classifier gave better accuracy on Devanagri, and Bangla numerals and gave average results for other languages.

3) Most of the published research studies propose a solution for one language or even a subset of a language.

4) It is observed that researchers are increasingly using Convolution Neural Networks (CNN) for the recognition of handwritten and machine-printed characters. This is due to the fact that CNN based architectures are well suited for recognition tasks where input is an image.

## 11.3 A COMPARATIVE STUDY OF TEXT COMPRESSION ALGORITHMS [9]

Data Compression is the science and art of representing information in a compact form. For decades, Data compression has been one of the critical enabling technologies for the ongoing digital multimedia revolution. There are lots of data compression algorithms which are available to compress files of different formats. Experimental results and comparisons of the lossless compression algorithms using Statistical compression techniques and Dictionary based compression techniques were performed on text data. Statistical coding techniques the algorithms such as Shannon-Fano Coding, Huffman coding, Adaptive Huffman coding, Run Length Encoding and Arithmetic coding are considered. Lempel Ziv scheme which is a dictionary based technique is divided into two families: those derived from LZ77 (LZ77, LZSS, LZH and LZB) and those derived from LZ78 (LZ78, LZW and LZFG).

The size of data is reduced by removing the excessive information. The goal of data compression is to represent a source in digital form with as few bits as possible while meeting the minimum requirement of reconstruction of the original. Data compression can be lossless, only if it is possible to exactly reconstruct the original data from the compressed version. Examples of such lossless data are medical images, text and images preserved for legal reason, some computer executable files, etc. Another family of compression algorithms is called lossy as these algorithms irreversibly remove some parts of data and only an approximation of the original data can be reconstructed. Multimedia images, video and audio are more easily compressed by lossy compression techniques. Lossy algorithms achieve better compression effectiveness than lossless algorithms, but lossy compression is limited to audio, images, and video, where some loss is acceptable. This session examines the performance of statistical compression techniques such as Shannon-Fano Coding, Huffman coding, Adaptive Huffman coding, Run Length Encoding and Arithmetic coding. The Dictionary based compression technique Lempel-Ziv scheme is divided into two families: those derived from LZ77 (LZ77, LZSS, LZH and LZB) and those derived from LZ78 (LZ78, LZW and LZFG).

**STATISTICAL COMPRESSION TECHNIQUES**

**a) SHANNON FANO CODING**

The algorithm is as follows:

Step 1. For a given list of symbols, develop a frequency or probability table.

Step 2. Sort the table according to the frequency, with the most frequently occurring symbol at the top.

Step 3. Divide the table into two halves with the total frequency count of the upper half being as close to the total frequency count of the bottom half as possible.

Step 4. Assign the upper half of the list a binary digit ‘0’ and the lower half a ‘1’.

Step 5. Recursively apply the steps 3 and 4 to each of the two halves, subdividing groups and adding bits to the codes until each symbol has become a corresponding leaf on the tree.

**b) HUFFMAN CODING**

The Huffman algorithm is simple and can be described in terms of creating a Huffman code tree. The procedure for building this tree is:

Step 1. Start with a list of free nodes, where each node corresponds to a symbol in the alphabet.

Step 2. Select two free nodes with the lowest weight from the list.

Step 3. Create a parent node for these two nodes selected and the weight is equal to the weight of the sum of two child nodes.

Step 4. Remove the two child nodes from the list and the parent node is added to the list of free nodes.

Step 5. Repeat the process starting from step-2 until only a single tree remains.

**c) ADAPTIVE HUFFMAN CODING**

The basic Huffman algorithm suffers from the drawback that to generate Huffman codes it requires the probability distribution of the input set which is often not available. The Adaptive Huffman coding technique was developed based on Huffman coding first by Newton Faller and by Robert G. Gallager and then improved by Donald Knuth and Jefferey S. Vitter. Both sender and receiver maintain dynamically changing Huffman code trees whose leaves represent characters seen so far. Initially the tree contains only the 0-node, a special node representing messages that have yet to be seen. Huffman tree includes a counter for each symbol and the counter is updated every time when a corresponding input symbol is coded. Huffman tree under construction is still a Huffman tree if it is ensured by checking whether the sibling property is retained. If the sibling property is violated, the tree has to be restructured to ensure this property. Storing Huffman tree along with the Huffman codes for symbols with the Huffman tree is not needed here. It is superior to Static Huffman coding in two aspects: It requires only one pass through the input and it adds little or no overhead to the output.

**d) ARITHMETIC CODING**

Huffman and Shannon-Fano coding techniques suffer from the fact that an integral value of bits is needed to code a character. Arithmetic coding completely bypasses the idea of replacing every input symbol with a codeword. Instead it replaces a stream of input symbols with a single floating point number as output. The basic concept of arithmetic coding was developed by Elias in the early 1960's and further developed largely by Pasco, Rissanen and Langdon. The main aim of Arithmetic coding is to assign an interval to each potential symbol. Then a decimal number is assigned to this interval. The algorithm starts with an interval of 0.0 and 1.0. After each input symbol from the alphabet is read, the interval is subdivided into a smaller interval in proportion to the input symbol's probability. This subinterval then becomes the new interval and is divided into parts according to probability of symbols from the input alphabet. This is repeated for each and every input symbol. And, at the end, any floating point number from the final interval uniquely determines the input data.



### e) LEMPEL ZIV ALGORITHMS

The Lempel Ziv Algorithm is an algorithm for lossless data compression. It is not a single algorithm, but a whole family of algorithms, stemming from the two algorithms proposed by Jacob Ziv and Abraham Lempel in their landmark papers in 1977 and 1978.

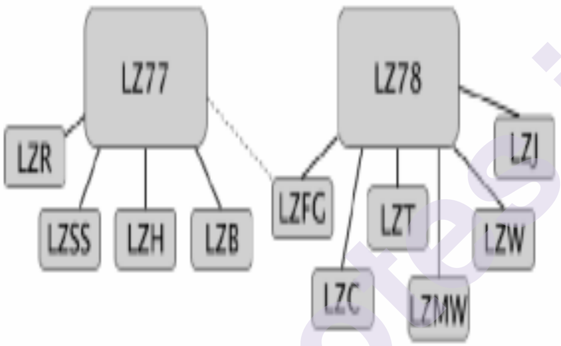



**Fig 16: LEMPEL ZIV ALGORITHMS**

Jacob Ziv and Abraham Lempel have presented their dictionary-based scheme in 1977 for lossless data compression. LZ77 exploits the fact that words and phrases within a text file are likely to be repeated. When there is repetition, they can be encoded as a pointer to an earlier occurrence, with the pointer accompanied by the number of characters to be matched. It is a very simple adaptive scheme that requires no prior knowledge of the source and seems to require no assumptions about the characteristics of the source.

The LZ77 algorithm is given below:

```
While (lookAheadBuffer not empty) {
get a reference (position, length) to longest match;
if (length > 0)
{
  output (position, length, next symbol);
  shift the window length+1 positions along;
  }
else {
output (0, 0, first symbol in the lookahead buffer);
shift the window 1 character along;
  }
}
```

### f) LZ78

In 1978 Jacob Ziv and Abraham Lempel presented their dictionary based scheme, which is known as LZ78. This dictionary has to be built both at the encoding and decoding side and they must follow the same rules to ensure that they use an identical dictionary. The codewords output by the algorithm consists of two elements where 'i' is an index referring to the longest matching dictionary entry and the first non-matching symbol. When a symbol that is not yet found in the dictionary, the codeword has the index value 0 and it is added to the dictionary as well. The algorithm gradually builds up a dictionary with this method. The algorithm for LZ78 is given below:

```
w := NIL;
while ( there is input ) {
  K := next symbol from input;
  if (wK exists in the dictionary) {
     w := wK;
  } else {
   output (index(w), K);
   add wK to the dictionary;
   w := NIL;
   }
}
```

LZ78 algorithm has the ability to capture patterns and hold them indefinitely but it also has a serious drawback. There are various methods to limit dictionary size, the easiest being to stop adding entries and continue like a static dictionary coder or to throw the dictionary away and start from scratch after a certain number of entries has been reached. The encoding done by LZ78 is fast, compared to LZ77, and that is the main advantage of dictionary based compression. The decompression in LZ78 is faster compared to the process of compression.

**EXPERIMENTAL RESULTS**

In this section we compare the performance of various Statistical compression techniques (Run Length Encoding, Shannon-Fano coding, Huffman coding, Adaptive Huffman coding and Arithmetic coding), LZ77 family algorithms (LZ77, LZSS, LZH and LZB) and LZ78 family algorithms (LZ78, LZW and LZFG). Research works done to evaluate the efficiency of any compression algorithm are carried out having two important parameters. Tested several times the practical performance of the above mentioned techniques on files of Canterbury corpus and have found out the results of various Statistical coding techniques and Lempel - Ziv techniques selected for this study.

**CONCLUSION**

Statistical compression techniques and Lempel Ziv algorithms were taken up to examine the performance in compression. In the Statistical compression techniques, Arithmetic coding technique outperforms the rest with an improvement of 1.15% over Adaptive Huffman coding, 2.28% over Huffman coding, 6.36% over Shannon-Fano coding and 35.06% over Run Length Encoding technique. LZB outperforms LZ77, LZSS and LZH to show a marked compression, which is 19.85% improvement over LZ77, 6.33% improvement over LZSS and 3.42% improvement over LZH, amongst the LZ77 family. LZFG shows a significant result in the average BPC compared to LZ78 and LZW. From the result it is evident that LZFG has outperformed the other two with an improvement of 32.16% over LZ78 and 41.02% over LZW.

## 11.4 PERFORMANCE EVALUATION IN CONTENT-BASED IMAGE RETRIEVAL: OVERVIEW AND PROPOSALS [10]


**Abstract**

Evaluation of retrieval performance is a crucial problem in content-based image retrieval (CBIR). Many different methods for measuring the performance of a system have been created and used by researchers. This article discusses the advantages and disadvantages of the performance measures currently used. Problems such as a common image database for performance comparisons and a means of getting relevance judgments (or ground truth) for queries are explained. The relationship between CBIR and information retrieval (IR) is made clear, since IR researchers have decades of experience with the evaluation problem. Many of their solutions can be used for CBIR, despite the differences between the fields. Several methods used in text retrieval are explained. Proposals for performance measures and means of developing a standard test suite for CBIR, similar to that used in IR at the annual Text REtrieval Conference (TREC), are presented.


**Introduction**

Early reports of the performance of CBIR systems were often restricted simply to printing the results of one or more example queries. This is easily tailored to give a positive impression, since developers can chooses queries which give good results. It is neither an objective performance measure, nor a means of comparing different systems. Many of the measures used in CBIR have long been used in IR. Several other standard IR tools have recently been imported into CBIR.

In the 1950s IR researchers were already discussing performance evaluation, and the first concrete steps were taken with the development of the SMART system in 1961. Other important steps towards common performance measures were made with the Craneld test. Finally, the TREC series started in 1992, combining many efforts to provide common performance tests. The TREC project provides a focus for these activities and is the worldwide standard in IR. Such novelties are included in TREC regularly.

**Information Retrieval**

Although performance evaluation in IR started in the 1950s, here we focus on newer results and especially on TREC and its achievements in the IR community. Not only did TREC provide an evaluation scheme accepted worldwide, but it also brought academic and commercial developers together and thus created a new dynamic for the field.

**Data Collections**

The TREC collection is the main collection used in IR. Co-sponsored by the National Institute of Standards and Technology and the Defense Advanced Research Projects Agency, TREC has been held annually since its inception. A large amount of training data is also provided before the conference. Special evaluations exist for interactive systems, spoken language, high-precision and cross-language retrieval. The collections can grow as computing power increases, and as new research areas are added.

**Relevance judgments**

The determination of relevant and non-relevant documents for a given query is one of the most important and time-consuming tasks. TREC uses the following working definition of relevance: If you were writing a report on the subject of the topic and would use the information contained in the document in the report, then the document is relevant. Only binary judgments are made, and a document is judged relevant if any piece of it is.

**Performance measures**

The most common evaluation measures used in IR are precision and recall, usually presented as a precision vs. recall graph. Researchers are familiar with PR graphs and can extract information from them without interpretation problems.

$$precision = \frac{\text{No. relevant documents retrieved}}{\text{Total No. documents retrieved}},$$

$$recall = \frac{\text{No. relevant documents retrieved}}{\text{Total No. relevant documents in the collection}}.$$

**Basic Problems in performance evaluation in CBIR**

The current status of performance evaluation in CBIR is far from that in IR. There are many different groups who are working with different sets of specialized images. There is neither a common image collection, nor a common way to get relevance judgments, nor a common evaluation scheme.

**Defining a common image collection**

There are several problems which must be addressed in order to create a common image collection. The greatest problem is to create a collection with enough diversity to cater for the diverse, partly specialized domains in CBIR such as medical images, car images, face recognition and consumer photographs.

A common means of constructing an image collection is to use Corel photo CDs, each of which usually contains 100 broadly similar images. Most research groups use only a subset of the collection, which can result in a collection consisting of several highly dissimilar groups of images, with relatively high within-group similarity. This can lead to great apparent improvements in performance: it is not too hard to distinguish sunsets from underwater images of fish! A good candidate for a standard collection could be the images and videos from MPEG-7.

An alternative approach is for CBIR researchers to develop their own collection. Such a project is underway at the the University of Washington in Seattle and is freely available without any copyright and owners annotated photographs of different regions and topics. It is still small (~500 images), but several groups are contributing to enlarge the data set. Collection size should be sufficiently high that the trade-of between speed and accuracy can be evaluated. In IR it is quite normal to have millions of documents whereas in CBIR most systems work with a few thousand images and some even with fewer than one hundred.

**Obtaining relevance judgments**

In CBIR there is not yet a common means of obtaining relevance judgments for queries. A very common technique is to use standard image databases with sets of different topics such as the Corel collection. Relevance “judgments" are given by the collection itself. Grouping is not always based on global visual similarity, but often on the contained objects. In some studies images which are too visually different are removed from the collection, which definitely improves results.

**Image grouping**

An alternative approach is for the collection creator or a domain expert to group images according to some criteria. Domain expert knowledge is very often used in medical CBIR. This can be seen as real groundtruth, because the images have a diagnosis certified by at least one medical doctor. These groups can then be used like the subsets discussed above.

**Simulating users**

Some studies simulate a user, by assuming that users' image similarity judgments are modeled by the metric used by the CBIR system, plus noise. Real users are very hard to model: Tversky (1977) has shown that human similarity judgments seem not to obey the requirements of a metric, and they are certainly user- and task-dependent. Such simulations cannot replace real user studies.

**Performance Evaluation Methods**

**User comparisons**

User comparison is an interactive method. It is hard to get a large number of such user comparisons as they are time-consuming. Users are given two or more different results and allowed to choose the one which is preferred or found to be most relevant to the query. This method needs a base system or another system for comparison.

**Single-valued measures**

Rank of the best match Berman & Shapiro (1999) measure whether the \most relevant" image is in either the first 50 or first 500 images retrieved. 50 represents the number of images returned on screen and 500 is an estimate of the maximum number of images a user might look at when browsing.

Error rate Hwang et al. (1999) use this measure, which is common in object or face recognition. It is in fact a single precision value, so it is important to know where the value is measured.

$$\text{Error rate} = \frac{\text{No. non-relevant images retrieved}}{\text{Total No. images retrieved}}.$$

**Retrieval efficiency**

Muller & Rigoll (1999) define Retrieval efficiency as specified below. If the number of images retrieved is lower than or equal to the number of relevant images, this value is the precision, otherwise it is the recall of a query. This definition can be misleading since it mixes two standard measures.

$$\text{Retrieval efficiency} = \begin{cases} \dfrac{\text{No. relevant images retrieved}}{\text{Total No. images retrieved}} & \text{if No. retrieved} \\ & > \text{No. relevant}, \\ \dfrac{\text{No. relevant images retrieved}}{\text{Total No. relevant images}} & \text{otherwise.} \end{cases}$$

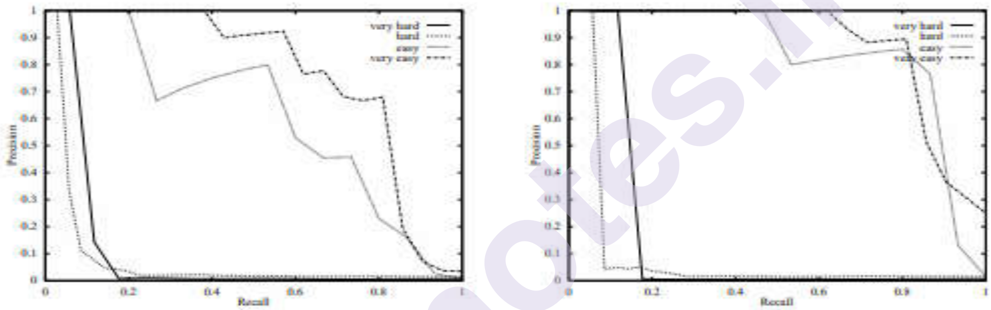
### Correct and incorrect detection

Ozer et al. (1999) use these measures in an object recognition context. The numbers of correct and incorrect classifications are counted. When divided by the number of retrieved images, these measures are equivalent to error rate and precision.

### Graphical representations

### Precision vs. recall graphs

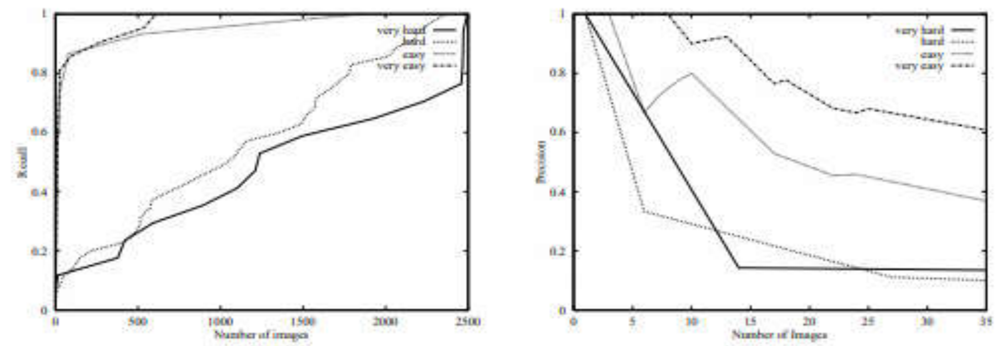
PR graphs are a standard evaluation method in IR and are increasingly used by the CBIR community. PR graphs contain a lot of information, and their long use means that they can be ready easily by many researchers. It is also common to present a partial PR graph (e.g. He (1997)). This can be useful in showing a region in more detail, but it can also be misleading since areas of poor performance can be omitted. Interpretation is also harder, since the scaling has to be watched carefully. A partial graph should always be used in conjunction with the complete graph.




(a) without feedback (b) with feedback

**Fig 17: PR graphs for four different queries both without and with feedback.**

Correctly retrieved vs. all retrieved graphs contain the same information as recall graphs, but differently scaled. Fraction correct vs. No. images retrieved graphs are equivalent to precision graphs. Average recognition rate vs. No. images retrieved graphs show the average percentage of relevant images among the first N retrievals. This is equivalent to the recall graph.




(a) recall graph (b) precision graph

Fig 18: Recall vs. No. of images graph and partial precision vs. No. of images graph

**CONCLUSIONS**

Current section gives an overview of existing performance evaluation measures in CBIR. The need for standardized evaluation measures is clear, since several measures are slight variations of the same definition. This makes it very hard to compare the performance of systems objectively. To overcome this problem a set of standard performance measures and a standard image database is needed. We have proposed such a set of measures, similar to those used in TREC. A frequently updated shared image database and the regular comparison of system performances would be of great benefit to the CBIR community.

## 11.5 REFERENCES

1. Digital Watermarking. https://www.techopedia.com/definition/24927/digital-watermarking

2. Rashid A. Digital watermarking applications and techniques: a brief review. International Journal of Computer Applications Technology and Research. 2016;5(3):147-50.

3. Digital Watermarking and its Types. https://www.geeksforgeeks.org/digital-watermarking-and-its-types/.

4. Jain S. Digital watermarking techniques: a case study in fingerprints & faces. InProc. Indian Conf. Computer Vision, Graphics, and Image Processing 2000 Dec (pp. 139-144).

5. Joshi, Manjunath & Joshi, Vaibhav & Raval, Mehul. (2013). Multilevel Semi-fragile Watermarking Technique for Improving Biometric Fingerprint System Security. Communications in Computer and Information Science. 276. 10.1007/978-3-642-37463-0_25.

6. Ganta S, Svsrk P. A novel method for Indian vehicle registration number plate detection and recognition using image processing techniques. Procedia Computer Science. 2020 Jan 1;167:2623-33.

7. Zhao ZQ, Zheng P, Xu ST, Wu X. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems. 2019 Jan 28;30(11):3212-32.

8. Chang SL, Hsu CC, Lu TC, Wang TH. Human body tracking based on discrete wavelet transform. InProceedings of the 2007 WSEAS International Conference on Circuits, Systems, Signal and Telecommunications 2007 Jan 17 (pp. 113-122).

9. Shanmugasundaram S, Lourdusamy R. A comparative study of text compression algorithms. International Journal of Wisdom Based Computing. 2011 Dec;1(3):68-76.

10. Müller H, Müller W, Squire DM, Marchand-Maillet S, Pun T. Performance evaluation in content-based image retrieval: overview and proposals. Pattern recognition letters. 2001 Apr 1;22(5):593-601.

## 11.6 MOOCS

1. Watermarking Basics. https://www.coursera.org/lecture/hardware-security/watermarking-basics-UHE3w.
2. Biometric Authentication. https://www.coursera.org/lecture/usable-security/biometric-authentication-RXVog.
3. Biometrics. https://www.udemy.com/course/biometrics/.
4. YOLO: Automatic License Plate Detection & Extract text App. https://www.udemy.com/course/deep-learning-web-app-project-number-plate-detection-ocr/
5. Object Detection. https://www.coursera.org/lecture/convolutional-neural-networks/object-detection-VgyWR.
6. Introduction to Optical Character Recognition. https://www.coursera.org/lecture/python-project/introduction-to-optical-character-recognition-n8be7.
7. Introduction to Data Compression. https://www.coursera.org/lecture/algorithms-part2/introduction-to-data-compression-OtmHU.

## 11.7 VIDEO LINKS

1. INTRODUCTION TO DIGITAL WATERMARKING. https://www.youtube.com/watch?v=WvRBKn8-JJA
2. Digital Watermarking – Introduction. https://www.youtube.com/watch?v=gd2W0vaKTxA
3. What is Biometric Authentication. https://www.youtube.com/watch?v=MBtzOzPakt8

4. Biometric authentication and its types and methods, information security. https://www.youtube.com/watch?v=tTnkq6Y3Hdg.
5. Vehicle License Plate Recognition. https://www.youtube.com/watch?v=CVDTtRiIXME
6. Vehicle Number Plate Recognition using MATLAB. https://www.youtube.com/watch?v=p_g-g7C3uHw.
7. Handwritten and Printed Text Recognition. https://www.youtube.com/watch?v=H64vHn_R0vg
8. OCR Explained...Handwriting Recognition!!!. https://www.youtube.com/watch?v=i_XJa165_9I.

❖❖❖❖

# Digital Image Restoration in Matlab: A Case Study on Inverse and Wiener Filtering

**Preprint** · February 2019

DOI: 10.20944/preprints201811.0566.v2



Mohammad Mahmudur Rahman Khan
Vanderbilt University


Shadman Sakib
Leading University


Md. Abu Bakr Siddique
International University of Business Agriculture and Technology

 doi:10.20944/preprints201811.0566.v2

# Digital Image Restoration in Matlab: A Case Study on Inverse and Wiener Filtering

Mohammad Mahmudur Rahman Khan[1*], Shadman Sakib[2#], Rezoana Bente Arif[2@], and Md. Abu Bakr Siddique[2$]
[1]*Dept. of ECE, Mississippi State University, Mississippi State, MS 39762, USA*
[2]*Dept. of EEE, International University of Business Agriculture and Technology, Bangladesh*
mrk303@msstate.edu[*], sakibshadman15@gmail.com[#], rezoana@iubat.edu[@], absiddique@iubat.edu[$]
Corresponding Author: absiddique@iubat.edu[$]

***Abstract*—In this paper, at first, a color image of a car is taken. Then the image is transformed into a grayscale image. After that, the motion blurring effect is applied to that image according to the image degradation model described in equation 3. The blurring effect can be controlled by a and b components of the model. Then random noise is added in the image via Matlab programming. Many methods can restore the noisy and motion blurred image; particularly in this paper Inverse filtering as well as Wiener filtering are implemented for the restoration purpose. Consequently, both motion blurred and noisy motion blurred images are restored via Inverse filtering as well as Wiener filtering techniques and the comparison is made among them.**

***Keywords*—Color image, grayscale image, motion blurring, random noise, inverse filtering, Wiener filtering, restoration of an image.**

## I. INTRODUCTION

In digital image processing, image restoration is an essential approach used for the retrieval of uncorrupted, original image from the blurred and noisy image [1, 2] because of motion blur, noise, etc. caused by environmental effects [3] and camera misfocus. Image blur may occur for many reasons such as motion blur which is due to the sluggish camera shutter speed comparative to the instantaneous motion of the targeted object [4]. The image also may subject to several forms of noises such as Poisson noise, Gaussian noise, etc. Poisson noise is controlled by signal and it is associated with the low light sources owing to photon counting statistic [4]. In contrast, the reason of Gaussian noise is because of electronic components and broadcast transmission effects [4]. In short, the term image restoration is an inverse process [5] by which the uncorrupted, original image can be recovered from the degraded form of the actual image [6]. There are many useful applications of digital image restoration in several fields including the area of astronomical imaging, medical imaging, media and filmography, security and surveillance videotapes, law enforcement and forensic science, image and video coding, centralized aviation assessment procedures [7], uniformly blurred television pictures restoration [8], etc. Several algorithmic techniques such as Artificial Neural Network [9], Convolutional neural Network [10], and K-nearest Neighbors [11] can also be applied in image processing techniques such as segmentation, thresholding and filtering. The technique used in image restoration is known as filtering which suppresses or removes unwanted components or features from the images. The most popular filtering techniques are used in image restoration in recent times are inverse filtering and Wiener filtering [12].

Inverse filter is a handy technique for image restoration if a proper degradation function can be modeled for the corrupted image. The performance of the inverse filter is quite right when the noise does not corrupt images, but in the presence of noise in the images, performance degrades significantly as high pass inverse filtration cannot eliminate noise properly because noise tends to be high frequency.

Wiener filter is incorporated with low pass filter together with high pass filter; as a result, it works actively in the existence of additive noise within the image. It performs deconvolution operation (high pass filtering) to invert motion blurring and also perform compression operation i.e. (low pass filtering) to eliminate the additive noise. Furthermore, in the process of inverting motion blurring and noise elimination, Wiener filter diminishes the overall mean square inaccuracy between the original and the output image of the filtration.

In this paper, the implementation of inverse filtering and Wiener filtering are analyzed for image restoration. Inverse filtering is applied into a motion blurred car image at first, and then wiener filtering is also used to the same image. After that, inverse and Wiener filtering are performed on the same motion blurred car image with additive noise. Finally, the comparison is made between inverse and Wiener filtering regarding their performances in restoring motion blurred images with and without additive noise.

## II. LITERATURE REVIEW

Over the past two decades, the technique of image processing has taken its place into every aspect of today's technological society. In digital image processing, there are a variety of essential steps involved such as image enhancement, pre-processing of images, image segmentation, image restoration and reconstruction of images etc. Among them, image restoration plays a vital role in today's world. It has several fields of applications in the areas of astronomy, remote sensing, microscopy, medical imaging, satellite imaging, molecular spectroscopy, law enforcement, and digital media restoration etc. Image restoration is very challenging as there is a lot of interference and noise in the environment like Gaussian noise, multiplicative noise, and impulse noise etc, inclusive of the camera such as wide angle lens, long exposure times, wind speed and degradation, blurring such as uniform blur, atmospheric blur, motion blur, and Gaussian blur etc. However, there are various methods of image restoration in the domain of image processing, for instance, Median filter, Wiener filtering, inverse filtering, Harmonic mean filter, Arithmetic mean filter, Max filter, and Maximum Likelihood (ML) method etc. Among these restoration methods, Wiener and inverse filtering method is the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 February 2019 doi:10.20944/preprints201811.0566.v2

simplest and advantageous method for overcoming the current restoration challenges mentioned above.

Stephen et al. outlined the restoration and reconstruction process from overlapping images of the multiple and same scene which is subjected to user-defined and data availability constraints on the support for the spatial domain process [13]. Michael et al. proposed an approach for out-of-focus blur and projector blur to reduce the image blur [14]. Yu et al. introduced an algorithm for the restoration of distorted and noisy images degraded by impulse and Gaussian noises [15]. Restoration of digitized photographs can be made by using multi-resolution texture synthesis and image imprinting [16]. Image restoration based on neural networks mainly focuses on spatial variation in terms of changeable regularization parameter for adaptively training the weights [17]. Moreover, the image can be restored by using a novel adaptive k-th nearest neighbor (KNN) strategy variant of the mean shift by knowing its neighbor [18], unsupervised, information-theoretic, adaptive filtering (UINTA) which improves the pixel intensities [19]. In another approach, the image can be restored from the mixed noise through minimization approach [20]. Recently, image restoration based on Convolutional Neural Network (CNN) achieved an encouraging implementation such as deep networks which performs non-local color image denoising [21], model-based optimization method to solve the various inverse problems like deblurring [22]. Moreover, the image can be restored by using the iterative method using denoising algorithm which provides a solution for the linear inverse problem [23].

The median filter is complex to execute as well as it's very time-consuming. Max filter cannot find the black or dark colored pixel of an image. When there is a need of sharp edges in the output, the arithmetic mean filter cannot provide the sharp edges rather it blurs the edges. ML method is sensitive to noise as the reversal of the imaging equation. Moreover, for pepper noise harmonic mean filter does not work well. After all those drawbacks of image restoration process mentioned above, Wiener and inverse filtering method are prominent and beneficial. The mean square error between the uncorrupted image is minimized by using the Wiener filter also it is not sensitive to noise. Inverse filtering is the prominent and simplest method to restore the image in the existence of noise and blur. In parallel, both wiener, and inverse filtering are used to retrieve the noisy and motion blurred images.

## III. FUNDAMENTALS OF IMAGE RESTORATION

Image restoration is a restoring or recovering process of a degraded image by utilizing some prior knowledge of degradation method which has degraded the image. So the image restoration process involves the estimation of the deteriorated model as well as the relevance of the inverse filtering to restore or retrieve the original image [24]. Although the reconstructed image may not be the exact form of the original image, it will be the approximation of the original image. Figure 1 below shows a fundamental model of image degradation and restoration procedure.

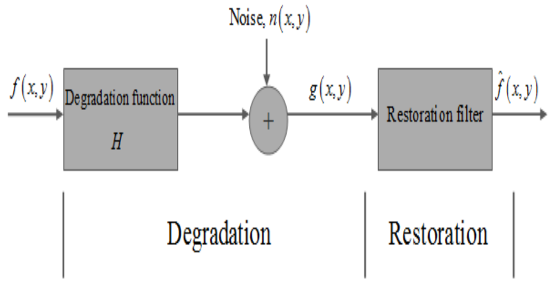


Fig. 1. The fundamental outline of image degradation and restoration procedure

In spatial domain, the degradation of the original image can be modeled as [25]:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \quad ...... (1)$$

Where,

$(x,y)$ = detached pixel coordinates of the image frame.

$f(x,y)$ = Original image

$g(x,y)$ = Degraded image

$h(x,y)$ = Image degradation function

$n(x,y)$ = Ad-on noise

As convolution operation within the spatial domain corresponds to multiplication in the frequency domain, the equation 1 can be rewritten as:

$$G(u,v) = H(u,v) \times F(u,v) + N(u,v) \quad ...... (2)$$

Now, Motion blur is present when there exists comparative motion in the midst of the recording device and the scene (object). However, the types of blur may be in the appearance of a translation, rotation, and scaling, or some combinations of these. Here only the critical case of a global translation will be considered.

Let's pretend the scene to be recorded interprets comparative to the camera at constant velocities a and b along the directions of x and y during the exposure time T. The frequency domain degradation function can be simplified as [26]:

$$H(u,v) = \frac{\sin(\pi(ua+vb)T)}{\pi(ua+vb)} e^{-j\pi(ua+vb)T} \quad ...... (3)$$

Image restoration process can be subdivided into two classes:

- Deterministic methods are applicable to images with a small amount of noise and a familiar degradation function.
- Stochastic techniques are to restore images according to some stochastic criterion.

*A. The Basics of Inverse Filtering*

Like any other unsupervised methods such as Fuzzy C-Means [27] and ADBSCAN [28] clustering, inverse filtering is also unsupervised. The basic image restoration model for inverse filtering is exposed in figure 2.

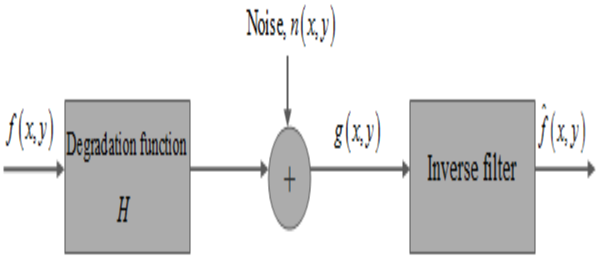



Fig. 2. Image restoration model (Inverse filtering)

When the degradation function H(u,v) is identified, the image can be returned to normal state by:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \;\ldots\ldots\; (4)$$

Now, in our case, we have added noise after implementing the motion blurring effect. Hence we have used the following formula to restore the original image:

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \;\ldots\ldots\; (5)$$

Since, the function of N(u,v) is random whose Fourier transform is generally unfamilier, it is impossible to retrieve F(u,v) accurately. The impact of noise is noteworthy for frequencies where H(u,v) has a tiny magnitude. In reality, H(u,v) usually decreases in size much more rapidly than N(u,v) and thus the noise effect N(u,v)/H(u,v) could take over the entire restoration result.

*B. The Basics of Wiener Filtering*

The basic image restoration model for Wiener filtering is modeled in figure 3.

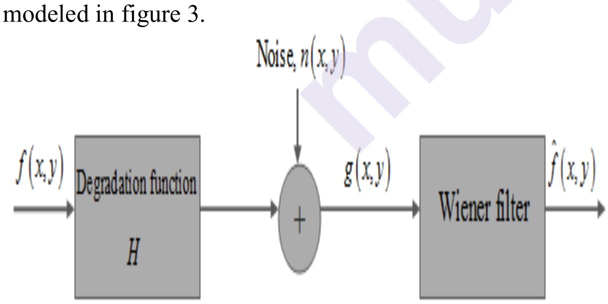



Fig. 3. Image restoration model (Wiener filtering)

Wiener filter exploits the previous knowledge of the spectral properties of the original signal and the noise and linear time-invariant rule to produce an output as close to the original image as feasible. In wiener filtering, it is presumed that the signal and noise are static linear stochastic processes with familiar spectral properties [29, 30]. Wiener filter tries to reconstruct the degraded image by minimizing an error function as designed by the following equation:

$$MSE = E[\{f(x,y) - \hat{f}(x,y)\}^2] \;\ldots\ldots\; (6)$$

Where,
$MSE$ = Mean square error
$E[.]$ = The expectation operation
$\hat{f}(x,y)$ = Restored image

The Wiener filter is to locate an approximation $\hat{f}(x,y)$ of the original image $f(x,y)$ so as to the mean square error between them is minimized. Wiener filter is represented as $L(u,v)$ as shown below [31, 32]:

$$L(u,v) = \frac{H^*(u,v)S_f(u,v)}{|H(u,v)|^2 S_f(u,v) + S_n(u,v)}$$
$$= \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}} \;\ldots\ldots\; (7)$$

Again,

$$L(u,v) = \frac{H^*(u,v)S_f(u,v)}{|H(u,v)|^2 S_f(u,v) + S_n(u,v)}$$
$$= \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \;\ldots\ldots\; (8)$$

Where,

$$K = \frac{S_n(u,v)}{S_f(u,v)}$$

$S_f(u,v)$ = Power spectrum of the original image

$S_n(u,v)$ = Noise power spectrum

Here, K is the inverse of SNR. The image and noise are considered as arbitrary processes. The Wiener filter can generate optimal estimate only if such stochastic processes are stationary Gaussian. These situations are not typically satisfied for real images. So the restored image can be expressed as:

$$\hat{f}(u,v) = L(u,v)G(u,v) \;\ldots\ldots\; (9)$$

## IV. RESULTS AND DISCUSSION

For this paper, the following color image of a car is used as made known in figure 4.

Fig. 4. The color image of a car

Then the image is transformed into a grayscale image in Matlab. The grayscale image is made known in figure 5 below.

Fig. 5. The grayscale image of figure 1

Now, the first task was to insert motion blur effect into the image according to equation (1). After doing so, the resulted image is represented in figure 6.

Fig. 6. Grayscale car image with motion blur effect

The effect of the motion blur can be controlled by a and b components of the model.

After applying the blur to the image inverse and, Wiener filterings are implemented to restore the image.

### A. The Results of Inverse Filtering

For inverse filtering, if we do not add any noise after the motion blur, then we can restore the same image before motion blur. The figure 7 below shows the effectiveness of inverse filtering without any noise.

Now, if we add some random noise to the image, then the filter performance degrades to some extent. The consequence of noise on the performance of inverse filtering is made known in figure 8. In figure 8, though the inverse filter is capable of inverting the effect of motion blur, it is not able to nullify the effect of noise. Here, a=0.0001 and b=0.1.




Fig. 7. Restoration of motion blurred car image by inverse filtering

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 February 2019 doi:10.20944/preprints201811.0566.v2

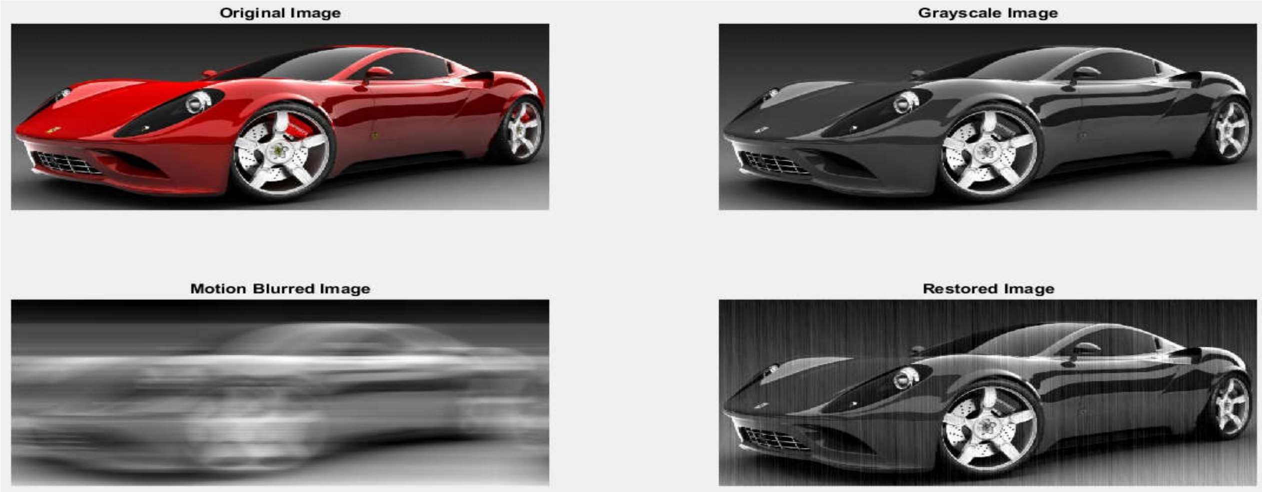



Fig. 8. Restoration of noisy motion blurred car image by inverse filtering

*B. The Results of Wiener Filtering*

The Wiener filter has a 'K' component which is inverse to the SNR. Now, if the noise power is zero, which means no noise, then the Wiener can restore the exact image which was corrupted by motion blur effect. In the following case, we have considered zero noise power, and figure 9 shows the performance of the Wiener filter.

Here, the restored image is almost exactly similar to the image before motion blur. However, if we change the k to 0.01, then there would be 1% noise added after the motion blur effect. If then we apply wiener filter, we will get the following result as represented in figure 10. It is observed that the Wiener filter is reversing the effect of motion blur, but still, there is some noise remaining in the picture.




Fig. 9. Restoration of motion blurred car image by Wiener filtering

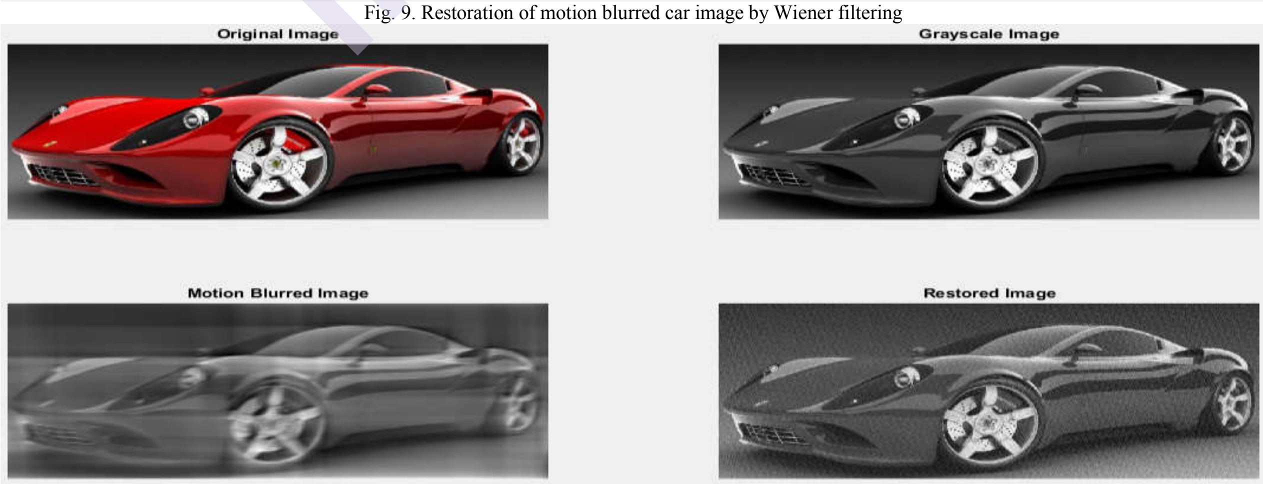



Fig. 10. Restoration of noisy motion blurred car image by Wiener filtering

## V. Conclusion

This paper presents inverse and wiener filterings' practical implementation on some images for image restoration. It is observed that both inverse and Wiener filtering work quite well in the absence of noise in restoring original image from its degraded version. But in the existence of additive noise wiener filtering works better for restoration purpose compared to inverse filtering. In subsequent works of this series, some other improved filtering techniques for image restoration will be discussed.

## References

[1] M. Trimeche, *et al.*, "Multichannel image deblurring of raw color components," in *Computational Imaging III*, 2005, pp. 169-179.

[2] L. Yang, "Image Restoration from a Single Blurred Photograph," in *Information Science and Control Engineering (ICISCE), 2016 3rd International Conference on*, 2016, pp. 405-409.

[3] T. F. Chan and J. J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods* vol. 94: Siam, 2005.

[4] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE signal processing magazine,* vol. 14, pp. 24-41, 1997.

[5] S. H. Lee, *et al.*, "Directional regularisation for constrained iterative image restoration," *Electronics letters,* vol. 39, p. 1642, 2003.

[6] A. Murli, *et al.*, "The wiener filter and regularization methods for image restoration problems," in *Image Analysis and Processing, 1999. Proceedings. International Conference on*, 1999, pp. 394-399.

[7] T. J. Kostas, *et al.*, "Super-exponential method for blur identification and image restoration," in *Visual Communications and Image Processing'94*, 1994, pp. 921-930.

[8] Z. Liu and J. Xiao, "Restoration of blurred TV picture caused by uniform linear motion," *Computer vision, graphics, and image processing,* vol. 43, p. 279, 1988.

[9] Md. Abu Bakr Siddique, *et al.*, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 118-123.

[10] Rezoana Bente Arif, *et al.*, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 112-117.

[11] Mohammad Mahmudur Rahman Khan, *et al.*, "Study and Observation of the Variation of Accuracies of KNN, SVM, LMNN, ENN Algorithms on Eleven Different Datasets from UCI Machine Learning Repository," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 124-129.

[12] N. Wiener, *et al.*, "Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications," 1949.

[13] S. E. Reichenbach and J. Li, "Restoration and reconstruction from overlapping images for multi-image fusion," *IEEE transactions on geoscience and remote sensing,* vol. 39, pp. 769-780, 2001.

[14] M. S. Brown, *et al.*, "Image pre-conditioning for out-of-focus projector blur," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 1956-1963.

[15] Y.-M. Huang, *et al.*, "Fast image restoration methods for impulse and Gaussian noises removal," *IEEE Signal Processing Letters,* vol. 16, pp. 457-460, 2009.

[16] H. Yamauchi and H.-P. Seidel, "Image restoration using multiresolution texture synthesis and image inpainting," in *null*, 2003, p. 120.

[17] S. W. Perry and L. Guan, "Weight assignment for adaptive image restoration by neural networks," *IEEE Transactions on neural networks,* vol. 11, pp. 156-170, 2000.

[18] C. V. Angelino, *et al.*, "Image restoration using a knn-variant of the mean-shift," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, 2008, pp. 573-576.

[19] S. P. Awate and R. T. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* pp. 364-376, 2006.

[20] Y. Xiao, *et al.*, "Restoration of images corrupted by mixed Gaussian-impulse noise via l1–l0 minimization," *Pattern Recognition,* vol. 44, pp. 1708-1720, 2011.

[21] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2017, pp. 3587-3596.

[22] K. Zhang, *et al.*, "Learning deep CNN denoiser prior for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Transactions on Image Processing,* 2018.

[24] R. C. Gonzalez and R. E. Woods, "Digital image processing," ed: Prentice hall New Jersey, 2002.

[25] D. Kundur and D. Hatzinakos, "A novel blind deconvolution scheme for image restoration using recursive filtering," *IEEE Transactions on Signal Processing,* vol. 46, pp. 375-390, 1998.

[26] R. C. Gonzalez, *et al.*, *Digital Image Publishing Using MATLAB*: Prentice Hall, 2004.

[27] M. Siddique, *et al.*, "Implementation of Fuzzy C-Means and Possibilistic C-Means Clustering Algorithms, Cluster Tendency Analysis and Cluster Validation," *arXiv preprint arXiv:1809.08417,* 2018.

[28] Mohammad Mahmudur Rahman Khan, *et al.*, "ADBSCAN: Adaptive Density-Based Spatial Clustering of Applications with Noise for Identifying Clusters with Varying Densities," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 107-111.

[29] R. G. Brown and P. Y. Hwang, "Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions," *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.,* 1997.

[30] R. Grover and P. Y. Hwang, "Introduction to random signals and applied Kalman filtering," *Willey, New York,* 1992.

[31] A. Khireddine, *et al.*, "Digital image restoration by Wiener filter in 2D case," *Advances in Engineering Software,* vol. 38, pp. 513-516, 2007.

[32] N. Kumar and K. K. Singh, "Wiener filter using digital image restoration," *Int. J. Electron. Eng.,* vol. 3, pp. 345-348, 2011.

# Restoring Degraded Face Images: A Case Study in Matching Faxed, Printed, and Scanned Photos

Thirimachos Bourlai, *Member, IEEE*, Arun Ross, *Senior Member, IEEE*, and Anil K. Jain, *Fellow, IEEE*

***Abstract*—We study the problem of restoring severely degraded face images such as images scanned from passport photos or images subjected to fax compression, downscaling, and printing. The purpose of this paper is to illustrate the complexity of face recognition in such realistic scenarios and to provide a viable solution to it. The contributions of this work are two-fold. First, a database of face images is assembled and used to illustrate the challenges associated with matching severely degraded face images. Second, a preprocessing scheme with low computational complexity is developed in order to eliminate the noise present in degraded images and restore their quality. An extensive experimental study is performed to establish that the proposed restoration scheme improves the quality of the ensuing face images while simultaneously improving the performance of face matching.**

***Index Terms*—Face recognition, faxed face images, image quality measures, image restoration, scanned face images.**

## I. Introduction

### A. Motivation

THE past decade has seen significant progress in the field of automated face recognition as is borne out by results of the 2006 Face Recognition Vendor Test (FRVT) organized by NIST [2]. For example, at a false accept rate (FAR) of 0.1%, the false reject rate (FRR) of the best performing face recognition system has decreased from 79% in 1993 to 1% in 2006. However, the problem of matching facial images that are severely degraded remains to be a challenge. Typical sources of image degradation include harsh ambient illumination conditions [3], low quality imaging devices, image compression, down sampling, out-of-focus acquisition, device or transmission noise, and motion blur [Fig. 1(a)–(f)]. Other types of degradation that have received very little attention in the face recognition literature include halftoning [Fig. 1(e)], dithering [Fig. 1(f)], and the presence of security watermarks on documents [Fig. 1(g)–(j)]. These types of degradation are observed in face images that are digitally acquired from printed or faxed documents. Thus, successful face recognition in the presence of such low quality probe images is an open research issue.

This work concerns itself with an automated face recognition scenario that involves comparing degraded facial photographs of subjects against their high-resolution counterparts (Fig. 2). The degradation considered in this work is a consequence of scanning, printing, or faxing face photos. The three types of degradation considered here are: 1) fax image compression,[1] 2) fax compression, followed by printing, and scanning, and 3) fax compression, followed by actual fax transmission, and scanning. These scenarios are encountered in situations where there is a need, for example, to identify legacy face photos acquired by a government agency that has been faxed to another agency. Other examples include matching scanned face images present in driver's licenses, refugee documents, and visas for the purpose of establishing or verifying a subject's identity.

The factors impacting the quality of degraded face photos can be 1) *person-related*, e.g., variations in hairstyle, expression, and pose of the individual; 2) *document-related*, e.g., lamination and security watermarks that are often embedded on passport photos, variations in image quality, tonality across the face, and color cast of the photographs; 3) *device-related*, e.g., the foibles of the scanner used to capture face images from documents, camera resolution, image file format, fax compression type, lighting artifacts, document photo size, and operator variability.

### B. Goals and Contributions

The goals of this work include 1) the design of an experiment to quantitatively illustrate the difficulty of matching degraded face photos against high-resolution images, and 2) the development of a preprocessing methodology that can "restore" the degraded photographs prior to comparing them against the gallery face images. In this regard, we first propose an iterative image restoration scheme. The objective functions employed to guide the restoration process are two image distortion metrics, viz., peak signal-to-noise ratio (PSNR) and the Universal Image Quality Index (UIQ) proposed by Wang and Bovik [5]. The target is to generate restored images that are of higher quality and that can achieve better recognition performance than their

Manuscript received March 08, 2010; revised January 10, 2011; accepted January 11, 2011. Date of publication February 04, 2011; date of current version May 18, 2011. This work was supported by the Center for Identification Technology Research (CITeR) at World Class University (WCU). The work of A. K. Jain was supported in part by the WCU program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-10008). A preliminary version of this work was presented at the First IEEE International Conference on Biometrics, Identity and Security (BIDS), September, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fabio Scotti.

T. Bourlai and A. Ross are with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: Thirimachos.Bourlai@mail.wvu.edu; Arun.Ross@mail.wvu.edu).

A. K. Jain is with the Computer Science and Engineering Department, Michigan State University, East Lansing, MI 48824 USA, and also with the Department of Brain and Cognitive Engineering, Korea University, Seoul 136-713, Korea (e-mail: jain@cse.msu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2109951

[1]In this work, *Fax image compression* is defined as the process where data (e.g., face images on a document) are transferred via a fax machine using the T.6 data compression, which is performed by the fax software.

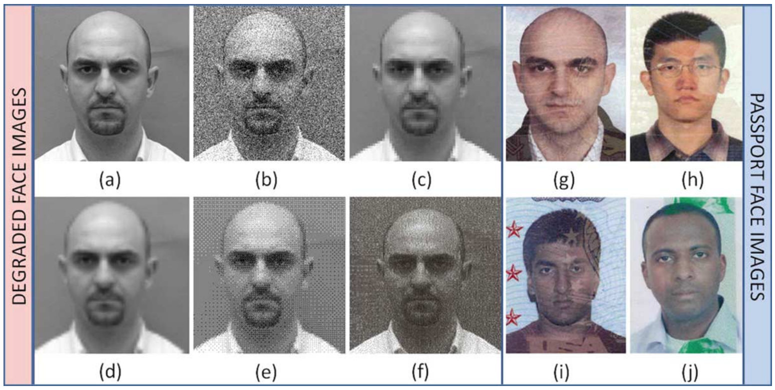



Fig. 1. Degraded face images: Low-resolution probe face images due to various degradation factors. (a) Original. (b) Additive Gaussian noise. (c) JPEG compressed (medium quality). (d) Resized to 10% and up-scaled to the original spatial resolution. (e) Half-toning. (f) *Floyd–Steinberg* dithering [4]. Mug-shots of face images taken from passports issued by different countries: (g) Greece (issued in 2006). (h) China (issued in 2008). (i) U.S. (issued in 2008). (j) Egypt (issued in 2005) [1].

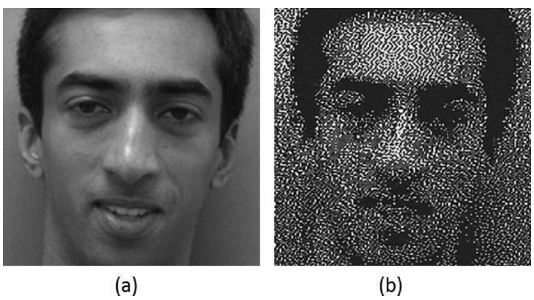



Fig. 2. Matching a high-resolution face image (a) against its degraded counterpart. The image in (b) is obtained by transmitting the document containing image (a) via a fax machine, and digitally scanning the resulting image.

original degraded counterparts. In order to facilitate this, a classification algorithm based on texture analysis and image quality is first used to determine the nature of the degradation present in the image. This information is then used to invoke the appropriate set of parameters for the restoration routine. This ensures that the computational complexity of both the classification and denoising algorithms is low, making the proposed technique suitable in real-time operations.

Second, we demonstrate that face recognition system performance improves when using the restored face image instead of the original degraded one. For this purpose, we perform *identification* tests on a variety of experimental scenarios, including 1) high-quality versus high-quality image comparison, and 2) high-quality versus degraded image comparison. In the high-quality versus high-quality tests, we seek to establish the baseline performance of each of the face recognition methods employed. In the high-quality versus degraded tests, we investigate the efficacy of matching the degraded face photographs (probe) against their high-resolution counterparts (gallery). Our approach avoids optimizing facial image representation purely for matching. Instead, the goal is to improve the quality of face images while at the same time boosting the matching performance. This can potentially assist human operators in verifying the validity of a match.

The key characteristics of the proposed face image restoration methodology are the following: 1) it can be applied on images impacted by various degradation factors (e.g., halftoning, dithering, watermarks, Gaussian noise, etc.); 2) individual images can have different levels of noise originating from a variety of sources; 3) the classification algorithm can automatically recognize the three main types of degradation studied in this paper; 4) it employs a combination of linear and nonlinear denoising methods (filtering and wavelets) whose parameters can be automatically adjusted to remove different levels of noise, and 5) the restoration process is computationally feasible (3 s per image in a Matlab environment) since parameter optimization is performed offline.



The proposed methodology is applicable to a wide range of face images—from high-quality raw images to severely degraded face images. To facilitate this study, initially a database containing passport photos and face images of 28 live subjects referred to as the WVU Passport Face Database was assembled. This dataset was extended to 408 subjects by using a subset of the FRGC2 [6] database. The purpose was to evaluate the restoration efficiency of our methodology in terms of identification performance on a larger dataset. Experiments were conducted using standard face recognition algorithms, viz., Local Binary Patterns [7], those implemented in the CSU Face Recognition Evaluation Project [8], and a commercial algorithm.

Section II briefly reviews related work in the literature. Section III presents the proposed restoration algorithm. Section V describes the technique used to evaluate the proposed algorithm. Section VI discusses the experiments conducted and Section VII provides concluding remarks.

## II. Background

The problem addressed in this paper is closely related to two general topics in the field of image processing: 1) image restoration and 2) super-resolution. The problem of restoring degraded images has been extensively studied [9]–[15]. However, most of the proposed techniques make implicit assumptions about the type of degradation present in the input image and do not necessarily deal with images whose degree of degradation is as severe as the images considered in this work. Furthermore, they do not address the specific problem of restoring *face* images where the goal is to simultaneously improve image quality and recognizability of the face. In the context of super-resolution, the authors in [16] and [17] addressed the problem of matching a high-spatial resolution gallery face image against a low-resolution probe. With the use of super-resolution methods [18], high-resolution images can be produced from either a single low-resolution image [19] or from a sequence of images [20]. While such techniques can compensate for disparity in image detail across image pairs, they cannot explicitly restore noisy or degraded content in an image. Also, when using a single low-resolution image to perform super-resolution, certain assumptions have to be made about the image structure and content.

The problem of matching passport photos was studied in [21] where the authors designed a Bayesian classifier for estimating the age difference between pairs of face images. Their focus was on addressing the age disparity between face images prior to matching them. However, their work did not address the specific problem of matching face images *scanned from documents* such as passports. Staroviodov *et al.* [22], [23] presented an automated system for matching face images scanned from documents against those directly obtained using a camera. The authors constrained their study to an earlier generation of passports (1990s) from a single country. Further, in the images considered in their work, the facial portion of the photograph was reasonably clear and not "contaminated" by any security marks. Therefore, the system's ability to automatically identify the face photograph was not severely compromised. To the best of our knowledge, the only work reported in the literature that addresses the problem of passport facial matching using *international* passports is [1].

## III. Face Image Restoration

Digital images acquired using cameras can be degraded due to many factors. Image denoising [24] is, therefore, a very important processing step to restore the structural and textural content of the image. While simple image filtering can remove specific frequency components of an image, it is not sufficient for restoring useful image content. For effective removal of noise and subsequent image restoration, a combination of linear denoising (using filtering), and nonlinear denoising (using thresholding) may be necessary in order to account for both noise removal as well as restoration of image features.

The quality of the denoiser used can be measured using the average mean square error $\mathrm{MSE}(\hat{h}, h_0) = E[(h_0 - \hat{h})^2]$, which is the error of the restored image $\hat{h}$ with respect to the true image $h_0$. Since the true image $h_0$ is unknown, the MSE corresponds to a theoretical measure of performance. In practice, this performance is estimated from denoising a single realization $h$ using different metrics such as the PSNR and/or UIQ:

- **Signal-to-Noise Ratio** (SNR): It is a measure of the magnitude of the signal compared to the strength of the noise. It is defined (in units of decibels) as:

$$\mathrm{SNR}(\hat{h}, h_0) = -20 \cdot \log_{10} \cdot \frac{\|h_0\|}{\left\|\hat{h} - h_0\right\|}. \tag{1}$$


Fig. 3. Denoising using a Wiener filter of increasing width $\gamma$.

This measure of performance requires knowledge of the true signal $h_0$ that might not be available in a real scenario. Thus, it should only be considered as an experimentation tool. Furthermore, this metric neglects global and composite errors, and in a practical scenario, its use is questionable. As a result, one should observe the image visually to judge the quality of the denoising method employed.

- **Peak Signal-to-Noise Ratio** (PSNR): This measure is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is defined (in units of decibels) via the MSE as follows:

$$\mathrm{PSNR}(\hat{h}, h_0) = 10 \cdot \log_{10} \cdot \frac{\mathrm{MAX}_i{}^2}{\mathrm{MSE}(\hat{h}, h_0)} \tag{2}$$

where $\mathrm{MAX}_i{}^2$ is the maximum fluctuation in the input image data type. For example, if the image has a double-precision floating-point data type, then $\mathrm{MAX}_i$ is 1, whereas in the case of an 8-bit unsigned integer data type, $\mathrm{MAX}_i$ is 255. A higher PSNR would normally indicate that the reconstruction is of higher quality. However, the authors in [5] illustrate some limitations of MSE/PSNR, and thus one must be very cautious in interpreting its outcome [25].

- **Universal Image Quality Index** (UIQ): The measure proposed in [5] was designed to model any image distortion via a combination of three main factors, viz., loss of correlation [(3): term 1], luminance distortion [(3): term 2], and contrast distortion [(3): term 3]. In our study, UIQ can be defined as follows: given a true image $x$ and a restored image $y$, let $\bar{x}, \bar{y}$ be the means, and $\sigma_x^2, \sigma_y^2$ be the variances of $x$ and $y$, respectively. Also, let $\sigma_{xy}$ be the covariance of $x$ and $y$. Then, UIQ can be denoted as follows:

$$\mathrm{UIQ} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 \sigma_y^2}. \tag{3}$$

### A. Linear and Nonlinear Denoising

*1) Image Filtering-Based Linear Denoising:* Linear methods can be used for image denoising so that the noise that perturbs an image is suppressed as much as possible. The filtering strength can be controlled by the filter width $\gamma$: higher values of $\gamma$ increase the blurring effect (see Fig. 3). When 2-D FIR filters are designed and used with the windowing method technique, $\gamma$ represents the window size of the digital filter in terms of pixels. In this paper, several smooth window functions were tested, viz., Hamming, Hanning, Bartlett, Blackman, boxcar, Kaiser, and Chebwin, with variable window sizes. Linear methods can cause image blurring. Therefore, these filters are efficient in denoising smooth images but not images with several discontinuities.

*2) Thresholding-Based Nonlinear Denoising:* When wavelets are used to deal with the problem of image denoising [24], the necessary steps involved are the following: 1) Apply discrete wavelet transform (DWT) to the noisy image by using a wavelet function (e.g., Daubuchies, Symlet, etc.). 2) Apply a thresholding estimator to the resulting coefficients thereby suppressing those coefficients smaller than a certain amplitude. 3) Reconstruct the denoised image from the estimated wavelet coefficients by applying the inverse discrete wavelet transform (IDWT).

The idea of using a thresholding estimator for denoising was systematically explored for the first time in [26]. An important consideration here is the choice of the thresholding estimator and threshold value used since they impact the effectiveness of denoising. Different estimators exist that are based on different threshold value quantization methods, viz., hard, soft, or semisoft thresholding.

Each estimator removes redundant coefficients using a nonlinear thresholding based on (4), where $h$ is the noisy observation, $\psi_m$ is the mother wavelet function, $m = (i, j)$ ($2^i$ is the scale and $j$ is the position of the wavelet basis), $\Omega$ is the thresholding estimator, $q$ is the thresholding type, and $T$ is the threshold used.

If $x$ is an input signal, then the estimators used in this paper are defined based on (5)–(7), where $\mu$ is a parameter greater than 1, and the superscripts $H$, $S$, and $SS$ denote hard, soft, and semisoft thresholding, respectively.

In nonlinear thresholding-based denoising methods [see (4)], translation invariance means that the basis $\{\psi_m = \psi_{i,j}\}_{i,j}$ is translation invariant $\forall m,\ \forall \tau \in \Phi$, where $\Phi$ is a lattice of $\Re^d$ and $d = 2$ for an image signal. While the Fourier basis is translation invariant, the orthogonal wavelet basis $\psi_m$ is not (in either the continuous or discrete settings)

$$\hat{h} = \sum_{|\langle h,\psi_m\rangle| > T} \langle h, \psi_m\rangle \psi_m = \sum_m \Omega_T^q(\langle h, \psi_m\rangle)\psi_m \quad (4)$$

$$\Omega_T^H(x) = \begin{cases} x, & \text{if } |x| > T \\ 0, & \text{if } |x| \le T \end{cases} \quad (5)$$

$$\Omega_T^S(x) = \begin{cases} \mathrm{sgn}(x)\cdot(|x - T|), & \text{if } |x| > T \\ 0, & \text{if } |x| \le T \end{cases} \quad (6)$$

$$\Omega_T^{SS}(x) = \begin{cases} 0, & \text{if } |x| \le T \\ x, & \text{if } |x| > \mu T \\ \mathrm{sgn}(x)\cdot\frac{|x-T|}{\mu-1}, & \text{if } |x| > T, \text{otherwise.} \end{cases} \quad (7)$$

Image denoising using the traditional orthogonal wavelet transforms may result in visual artifacts. Some of these can be attributed to the lack of translation invariance of the wavelet basis. One method to suppress such artifacts is to "average out" the translation dependence, i.e., through "cycle spinning" as proposed by Coifman [27]

$$\Theta_{\mathrm{TI}}(h) = \frac{1}{|\Phi|}\cdot\sum_{\tau\in\Phi}\Theta(h_\tau)_{-\tau} \quad (8)$$

where $\forall \tau \in \Phi$, $\Theta_{\mathrm{TI}}(h) = \Theta_{\mathrm{TI}}(h_\tau)_{-\tau}$. This is called **cycle spinning** denoising. If we have an $N$-sample data, then pixel precision translation invariance is achieved by having $N$ wavelet translation transforms (vectors) or $|\Phi| = N$.

Similar to cycle spinning denoising, thresholding-based translation invariant denoising can be defined as

$$\Theta_{\mathrm{TI}}(h) = \frac{1}{|\Phi|}\cdot\sum_{m,\tau\in\Phi}\Omega_T^q(\langle h, (\psi_m)_\tau\rangle)(\psi_m)_\tau. \quad (9)$$

The benefit of translation invariance over orthogonal thresholding is the SNR improvement afforded by the former. The problem with orthogonal thresholding is that it introduces oscillating artifacts that occur at random locations when $\tau$ changes. However, translation invariance significantly reduces these artifacts by the averaging process. A further improvement in SNR can be obtained by proper selection of the thresholding estimator.

## IV. Face Image Restoration Methodology

The proposed restoration methodology is composed of an *online* and an *offline* process (see Fig. 4). The *online process* has two steps. First, each input face image is automatically classified into one of three degradation categories considered in this work: 1) class 1: fax compression, 2) class 2: fax compression, followed by printing and scanning, and 3) class 3: fax compression, followed by fax transmission and scanning. In actual implementation, the system will not know whether the input face image is degraded or not. If the input face image is the original (good quality) image, it is assigned a fourth category, i.e., 4) class 4: good quality. Based on this classification, a restoration algorithm with a predefined meta-parameter set associated with the nature of degradation of the input image, is invoked. Each meta-parameter set is deduced during the *offline process*.

### A. Offline Process

Noniterative denoising methods (as those described above, viz., filtering and wavelet denoising with thresholding) derive a solution through an explicit numerical manipulation applied directly to the image in a single step. The advantages of noniterative methods are primarily ease of implementation and faster computation. Unfortunately, noise amplification is hard to control. Thus, when applied to degraded face images, they do not result in an acceptable solution. However, when they are applied iteratively and evaluated through a quality metric-based objective function, image reconstruction can be performed by optimizing this function. In our study, we employ such a scheme. At each step the system meta-parameters, i.e., 2-D FIR filter type/size, wavelet/thresholding type, and thresholding level, change incrementally within a predefined interval until the image quality of the reconstructed image is optimized in terms of some image distortion metric.

Mathematically, this can be expressed as follows. In each iteration $i$ of the algorithm, let $\hat{h}$ be the noisy observation of a true 2-D image $h_0$, and $h_{\mathrm{l}}(p_{\mathrm{l}})$, $h_{\mathrm{nl}}(p_{\mathrm{nl}})$ be the estimated image after applying linear and nonlinear denoising, respectively,

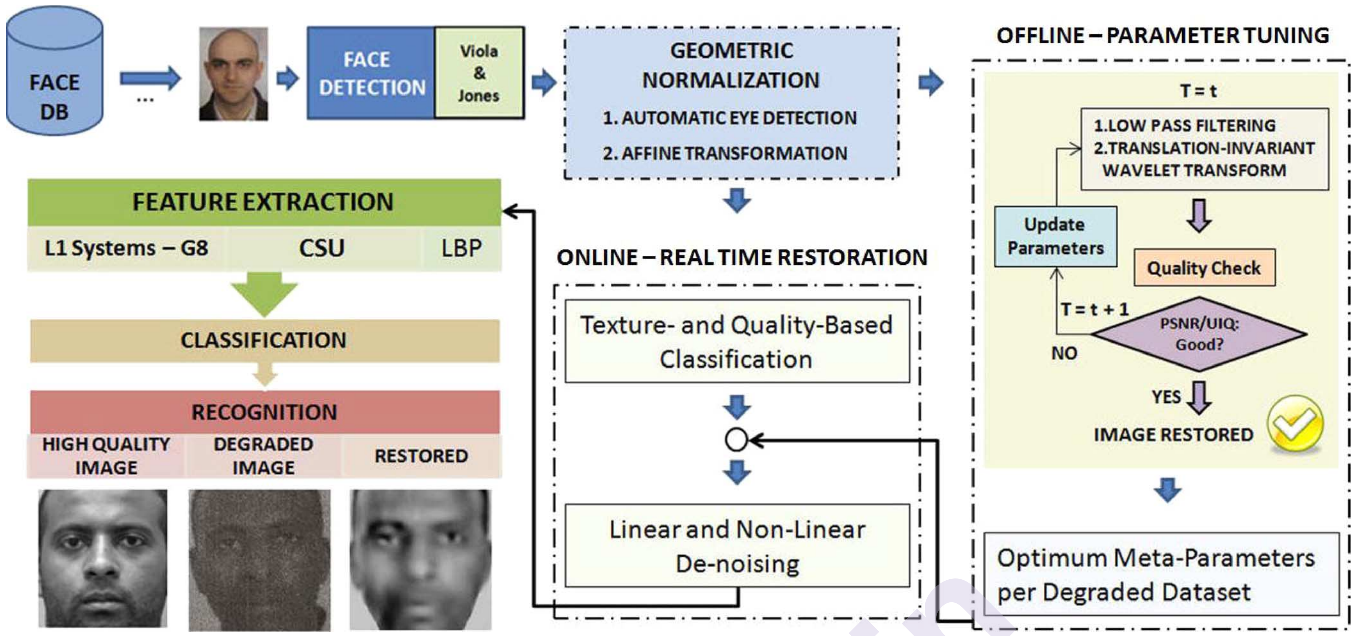



Fig. 4. Overview of the face image restoration methodology.

where $p_{\mathrm{l}} = \{f, \gamma\}$ denotes the set of linear denoising parameters, i.e., filter type and window size, and $p_{\mathrm{nl}} = \{w, q, T\}$ is the set of nonlinear parameters, i.e., wavelet type, thresholding type and level, respectively. Then, $\forall\ h \in \Xi$, where $\Xi$ represents a dataset of $N$ degraded images, given a finite domain $D = \{p_{\mathrm{l}}, p_{\mathrm{nl}}\}$ that represents the parameters employed (discrete or real numbers) and a quality metric function $Q$ such that $Q : D \rightarrow R$, the proposed reconstruction method works by finding the parameter set $\overline{p}$ in $D$ that maximizes $Q$

$$\begin{aligned}\overline{p} = \arg \max_{\{p_{\mathrm{l}}, p_{\mathrm{nl}}\} \in D} \{ & Q_i^{\mathrm{l}}[h_i^c, h_i^{\mathrm{l}}(p_{\mathrm{l}})], \\ & \text{or } Q_i^{\mathrm{nl}}[h_i^c, h_i^{\mathrm{nl}}(p_{\mathrm{nl}})], \\ & \text{or } Q_i^{\mathrm{lnl}}[h_i^c, h_i^{\mathrm{nl}}(h_i^{\mathrm{l}}(p_{\mathrm{l}}, p_{\mathrm{nl}}))]\}\end{aligned}$$

where the terms involved correspond to filtering (noted as $\mathrm{l}$), nonlinear denoising (noted as $\mathrm{nl}$), and their combination (noted as $\mathrm{lnl}$).

This procedure is iterated until convergence (i.e., stability of the maximum quality) by altering the constrained parameters (window/wavelet/thresholding type) and updating the window size and threshold level in an incremental way. The maximum number of iterations is empirically set. For instance, a threshold value of more than 60 results in removing too much information content. The application of this process to a degraded training dataset results in an estimated parameter set for each image. The optimum meta-parameter set for each degraded training dataset is obtained by averaging. The derived meta-parameter sets are utilized in the online restoration process.

### B. Online Process

In the online process (see Fig. 4), the degradation type of each input image is recognized by using a texture- and quality-based classification algorithm. First, the classifier utilizes the gray-tone spatial-dependence matrix, or cooccurrence matrix (COM) [28], which is the statistical relationship of a pixel's intensity to the intensity of its neighboring pixels. The COM measures the probability that a pixel of a particular gray level occurs at a specified direction and distance from its neighboring pixels. In this study, the main textural features extracted are inertia, correlation, energy, and homogeneity:

- **Inertia** is a measure of local variation in an image. A high inertia value indicates a high degree of local variation.
- **Correlation** measures the joint probability occurrence of the specified pixel pairs.
- **Energy** provides the sum of squared elements in the COM.
- **Homogeneity** measures the closeness of the distribution of elements in the COM to the COM diagonal.

These features are calculated from the cooccurrence matrix where pairs of pixels separated by a distance ranging from 1 to 40 in the horizontal direction are considered resulting in a total of 160 features per image (4 main textural features at 40 different offsets).

Apart from these textural features, image graininess is used as an additional image quality feature. Graininess is measured by the percentage change in image contrast of the original image before and after blurring is applied.[2] The identification of the degradation type of an input image is done by using the $k$-Nearest Neighbor ($k$-NN) method [29], [30] with $k = 5$. The online process restores the input image by employing the associated meta-parameter set (deduced in the offline process).

### C. Computation Time

The *online restoration process* when using MATLAB on a Windows Vista 32-bit system with 4-GB RAM and Intel Core Duo CPU T9300 at 2.5 GHz, requires about 0.08 s for the $k$-NN classification and about 2.5 s for image denoising, i.e., a total time of less than 3 s per image.

[2]Available: http://www.clear.rice.edu/elec301/Projects02/artSpy/graininess.html

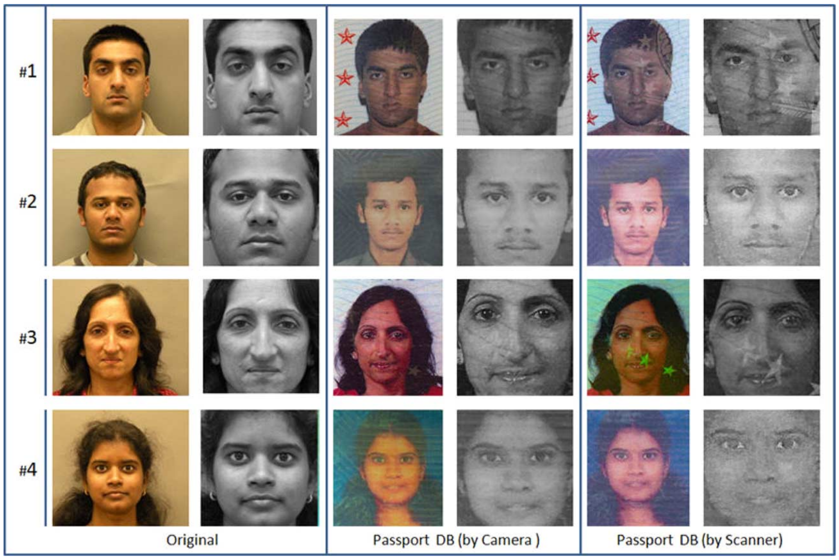



Fig. 5. Sample images of subjects in the three datasets of PassportDB.

## V. Degraded Face Image Databases

In this section, we will describe the hardware used for 1) the acquisition of the high-quality face images, and 2) for printing, scanning, and faxing the face images (along with the associated software). We will also describe the live subject-capture setup used during the data collection process and the three degraded face image databases used in this paper.

*1) Hardware and Subject-Capture Setup:* A NIKON Coolpix P-80 digital camera was used for the acquisition of the high-quality face images ($3648 \times 2736$) and an HP Office jet Pro L7780 system was used for printing and scanning images. The fax machine used was a Konica Minolta bizhub 501, in which the fax resolution was set to $600 \times 600$ dpi, the data compression method was MH/MR/MMR/JBIG, and transmission standard used for the fax communication line was super G3. The *Essential Fax* software was used to convert the scanned document of the initial nondegraded face photos into a PDF document with the fax resolution set to $203 \times 196$ dpi.

Our live subject-capture setup was based on the one suggested by the U.S. State Department, Bureau of Consular Affairs [31]. For the passport-capture setup we used the P-80 camera and the L7780 system. We acquired data from 28 subjects bearing passports from different countries, i.e., 4 from Europe, 14 from the United States, 5 from India, 2 from Middle East, and 3 from China; the age distribution of these participants was as follows: 20–25 (12 subjects), 25–35 (10 subjects), and over 35 (6 subjects). The database was collected over 2 sessions spanning approximately 10 days. In the beginning of the first session, the subjects were briefed about the data collection process after which they signed a consent document. During data collection, each subject was asked to sit $\sim$4 feet away from the camera. The data collection process resulted in the generation of three datasets, i.e., the NIKON Face Dataset (NFaceD) containing high-resolution face photographs from live subjects, the NIKON Passport Face Dataset (NPassFaceD) containing images of passport photos, and the HP Scanned Passport Face Dataset (HPassFaceD) containing face images scanned from the photo page of passports (see Fig. 5).

*2) Experimental Protocol:* Three databases were used in this paper (Fig. 6).

*Passport Database:* As stated above, the data collection process resulted in the generation of the *Passport Database (PassportDB)* composed of three datasets: 1) the NFaceD dataset that contains high-resolution face photographs from live subjects, 2) the NPassFaceD dataset that contains passport face images of the subjects acquired by using the P-80 camera, and 3) the HPassFaceD dataset that contains the passport face images of the subjects acquired by using the scanning mode of the L7780 machine.

In the case of NPassFaceD, three samples of the photo page of the passport were acquired for each subject. In the case of HPassFaceD, one scan (per subject) was sufficient to capture a reasonable quality mug-shot from the passport (Fig. 5).

*Passport-Fax Database:* This database was created from the *Passport Database* (Fig. 7). First, images in the Passport database were passed through four fax-related degradation scenarios. This resulted in the generation of four fax-passport datasets that demonstrate the different degradation stages of the faxing process when applied to the original passport photos: – **Dataset 1**: Each face image in the NPassFaceD/HPassFaceD datasets was placed in a Microsoft PowerPoint document. This document was then processed by the fax software producing a multipage PDF document with fax compressed face images. Each page of the document was then resized to $+150\%$. Then, each face image was captured at a resolution of $600 \times 600$ dpi by using a screen capture utility software (SnagIt v8.2.3). – **Dataset 2**: Same as *Dataset 1*, but this time each page of the PowerPoint document was resized to $+100\%$. Then each face image was captured at a resolution of $400 \times 400$ dpi. The purpose of employing this scenario was to study the effect of lower resolution of the passport face images on system performance. – **Dataset 3**: Following the same initial steps of *Dataset 1*, a multipage PDF document was produced with degraded images

| DATABASES | | | |
|---|---|---|---|
| | DATASETS | DESCRIPTION | Number of [Subjects / Samples per Subject] |
| PASSPORT | NFaceD | Controlled Still Face Images | 28 / 14 |
| | NFacePassD | Passport Images (Nikon-Camera) | 28 / 3 |
| | HFacePassD | Passport Images (HP-Scanner) | 28 / 1 |
| FRGC2 | EXP 1 | Controlled Still Face Images | 380 / 8 |
| BOTH - FAXED | D1 | • FAX Compression<br>- Face Captured at 600x600 dpi | 408 / 8 |
| | D2 | • FAX Compression<br>- Face Captured at 400x400 dpi | |
| | D3 | • FAX Compression<br>• Print<br>• Scan<br>- Face Captured at 600x600 dpi | |
| | D4 | • FAX Compression<br>• SENT via FAX<br>• Scan<br>- Face Captured at 600x600 dpi | |

Fig. 6. Description of the experimental protocol.

due to fax compression. The document was then printed and scanned at a resolution of $600 \times 600$ dpi. **– Dataset 4**: Again, we followed the same initial steps of *Dataset 1*. In this case, the PDF document produced was sent via an actual fax machine and each of the resulting faxed pages was then scanned at a resolution of $600 \times 600$ dpi.

*FRGC2-Passport FAX Database:* The primary goal of the *Face Recognition Grand Challenge* (FRGC) Database project was to evaluate the face recognition technology. In this work, we combined the FRGC dataset that has 380 subjects with our NFacePass dataset that consists of another 28 subjects. The extended dataset is composed of 408 subjects with eight samples per subject, i.e., 3264 high-quality facial images. The purpose was to create a larger dataset of high-quality face images that can be used to evaluate the restoration efficiency of our methodology in terms of identification performance, i.e., to investigate whether the restored face images can be matched with the correct identity in the augmented database. Following the process described for the previous database, four datasets were created and used in our experiments.

### *A. Face Image Matching Methodology*

The salient stages of the proposed method are described below:

1) **Face Detection**: The Viola & Jones face detection algorithm [32] is used to localize the spatial extent of the face and determine its boundary.
2) **Channel Selection**: The images are acquired in the RGB color domain. Empirically, it was determined that in the majority of passports, the Green channel (RGB color space) and the Value channel (HSV color space) are less sensitive to the effects of watermarking and reflections from the lamination. These two channels are selected and then added, resulting in a new single-channel image. This step is beneficial when using the Passport data. With the fax data this step is not employed since the color images are converted to grayscale by the faxing process.
3) **Normalization**: In the next step, a geometric normalization scheme is applied to the original and degraded images after detection. The normalization scheme compensates for slight perturbations in the frontal pose. *Geometric normalization* is composed of two main steps: eye detection and affine transformation. Eye detection is based on a template matching algorithm. Initially, the algorithm creates a *global eye* from all subjects in the training set and then uses it for eye detection based on a cross correlation score between the global and the test image. Based on the eye coordinates obtained by eye detection, the canonical faces are constructed by applying an affine transformation as shown in Fig. 4. These faces are warped to a size of $300 \times 300$. The *photometric normalization* applied to the passport images before restoration is a combination of homomorphic filtering and histogram equalization. The same process is used for the fax compressed images before they are sent to the fax machine.
4) **Image Restoration**: The methodology discussed in Section IV is used. By employing this algorithm, we process the datasets described in Section V and create their reconstructed versions that are later used for quality evaluation and identity authentication. Fig. 8 illustrates the effect of applying the restoration algorithm on some of the Passport Datasets (1, 3, and 4), i.e., passport faces a) subjected to T.6 compression (FAX SW) and restored; b) subjected to T.6 compression, printed, scanned, and restored; and c) subjected to T.6 compression, sent via fax machine, then scanned and finally restored. Note that in Fig. 8, the degraded faces in the left column are the images obtained after face detection and before normalization.
5) **Face Recognition Systems**: Both commercial and academic software were employed to perform the face recognition experiments: 1) Commercial software *Identity*

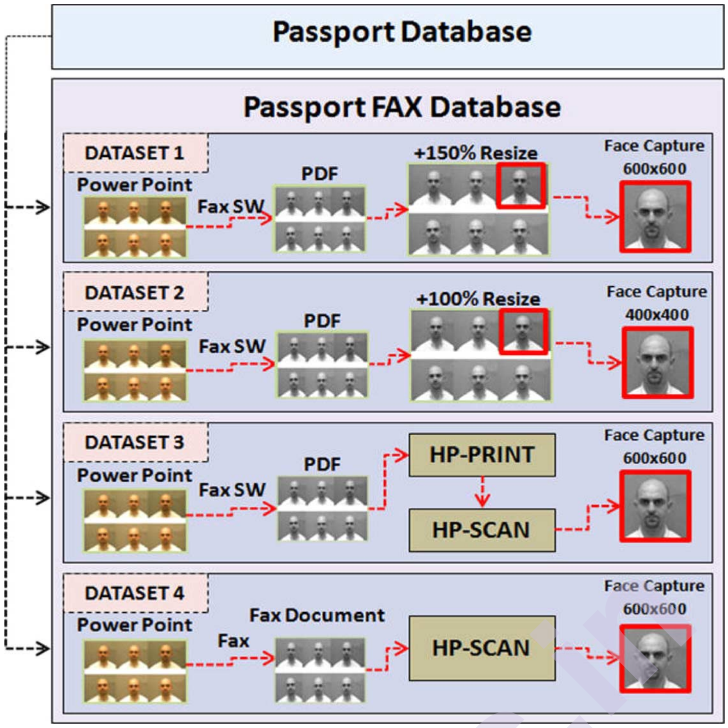



Fig. 7. Overview of the generation of the Passport FAX Database.

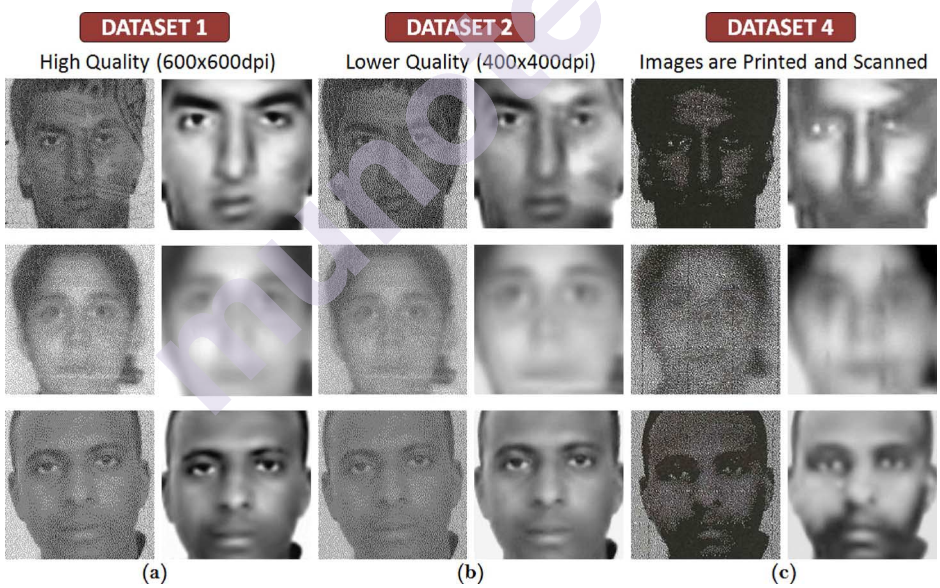



Fig. 8. Illustration of the effect of the proposed restoration algorithm. The input consists of (a) images subjected to fax compression and then captured at $600 \times 600$ dpi resolution; (b) images subjected to fax compression and then captured at $400 \times 400$ dpi resolution; (c) images subjected to fax compression then printed and scanned.



*Tools G8* provided by L1 Systems;[3] 2) standard face recognition methods provided by the CSU Face Identification Evaluation System [8], including *Principle Components Analysis* (PCA) [33]–[35], *a combined Principle Components Analysis and Linear Discriminant Analysis algorithm* (PCA+LDA) [36], the *Bayesian Intrapersonal/Extra-personal Classifier* (BIC) using either the Maximum likelihood (ML) or the Maximum *a posteriori* (MAP) hypothesis [37] and the *Elastic Bunch Graph Matching* (EBGM) method [38]; and (3) *Local Binary Pattern* (LBP) method [39].

[3]Available: www.l1id.com

## VI. Empirical Evaluation

The experimental scenarios investigated in this paper are the following: 1) evaluation of image restoration in terms of image quality metrics; 2) evaluation of the texture and quality based classification scheme; and 3) identification performance before and after image restoration.

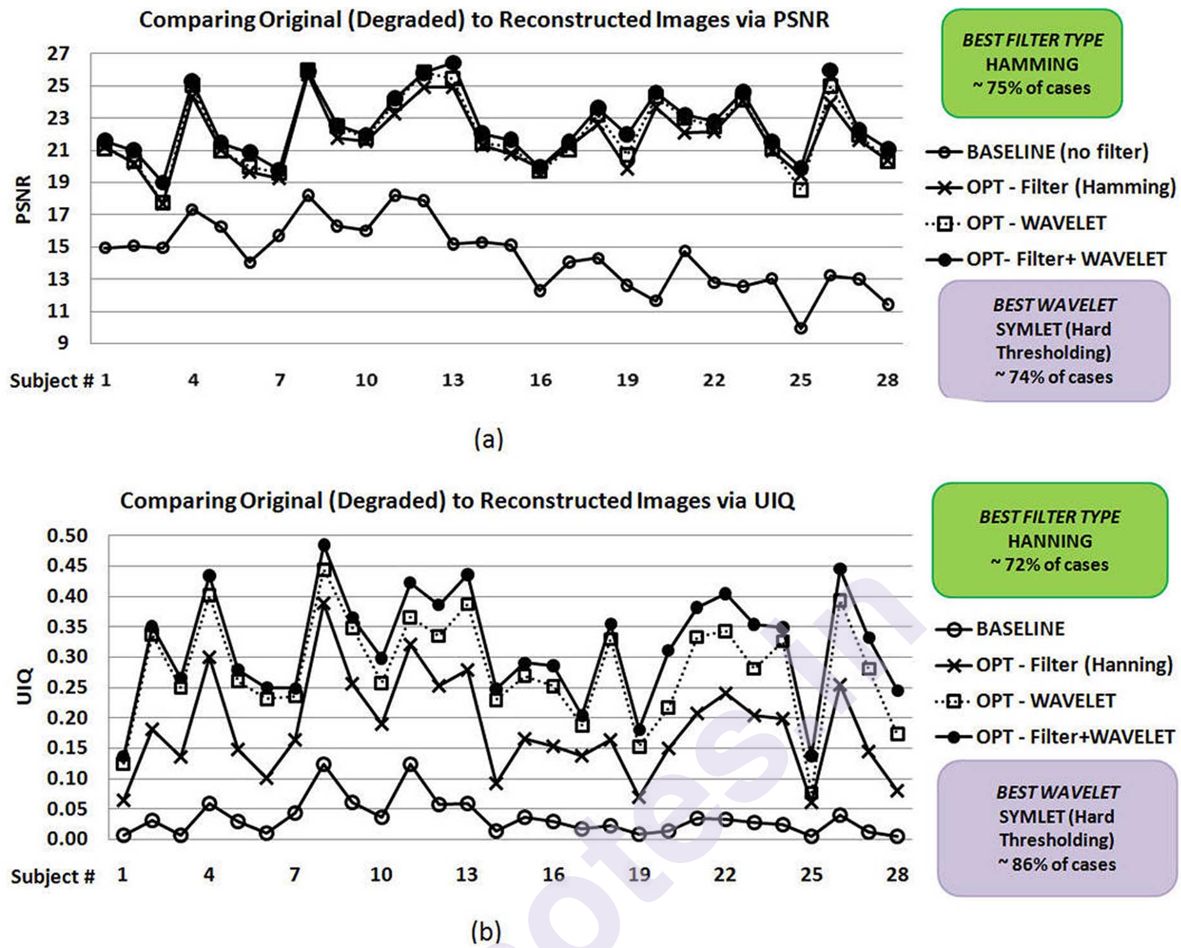



Fig. 9. Improvement in image quality as assessed by the PSNR and UIQ metrics. These metrics are computed by using the high-quality counterpart of each image as the "clean image."



### *A. Image Restoration Evaluation*

In this experiment, we demonstrate that the combination of filtering and TI-denoising is essential for improving the quality of restoration. Due to the absence of the ground truth passport data (digital version of the face images before they are printed and placed on the passport), we compare the high-quality live face images of each subject against their degraded version (due to fax compression) in terms of the PSNR and UIQ metrics. We investigate whether 1) linear filtering (2-D finite impulse response (FIR) filters that used the windowing method), 2) denoising, or 3) their combination is a favorable choice for restoration.

In the *first experiment*, we tested seven windows, i.e., boxcar, Hamming, Hanning, Bartlett, Blackman, Kaiser, and Chebwin, and varied the window size from 3 to 60 in increments of 2. When PSNR is used, in the majority of the cases (~75%), the most efficient window for image restoration was Hamming. This is illustrated in Fig. 9(a). The same trend in results is observed when using the UIQ metric; however, in a majority of the cases (~72%), the most efficient window for image restoration was Hanning [Fig. 9(b)]. The main conclusion from this experiment is that image filtering does improve the quality of the degraded fax images.

In the *second experiment*, we determined the TI-wavelet parameter set that could offer the best tradeoff between image restoration (in terms of PSNR and UIQ) and computational complexity. Thus, in this experiment, we examined the use of different filters (Daubechies and Symlets), thresholding type (hard, soft, semisoft), and level of thresholding (from 5 to 75 in increments of 5). The *Daubechies* filters are minimal phase filters that generate wavelets which have a minimal support for a given number of vanishing moments. *Symlets* are also wavelets within a minimum size support for a given number of vanishing moments. However, they are as symmetrical as possible in contrast to the Daubechies filters which are highly asymmetrical. Experimental results show that in terms of the *average metric* (PSNR/UIQ) for all subjects, the best option is to employ Symlet wavelets with hard thresholding [see Fig. 9(a), (b)].

In the *third experiment*, we investigate the effect of combining filtering with denoising. Fig. 9(a) and (b) shows that, overall, the best option is to combine Hanning Filtering with Symlets (hard thresholding). We can see that in the baseline case, the quality of the degraded fax images before restoration is very low (almost zero in some cases). By using filtering, denoising, or both and employing the proposed iterative approach, the average image quality significantly improves. In the "best"

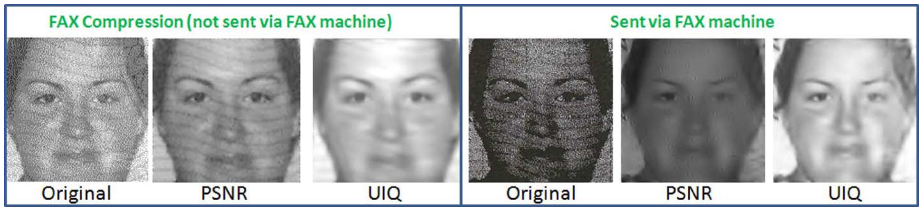



Fig. 10. Comparison of degraded images and their reconstructed counterparts after employing the proposed restoration method using PSNR/UIQ as quality metrics. UIQ appears to result in, at least visually, better images.

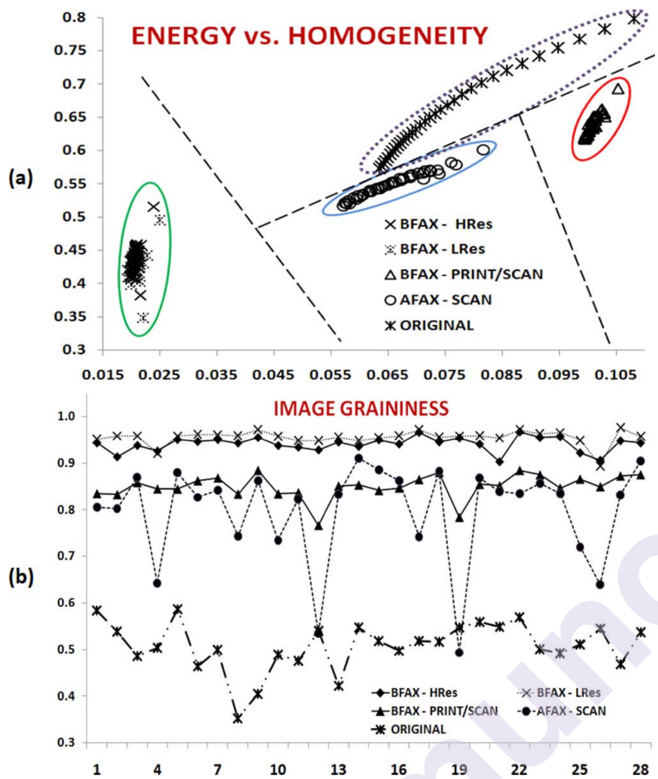



Fig. 11. (a) Clustering results when using textural features. (b) Importance of graininess in identifying the degraded datasets. B4FAX = Fax Compression (not sent via Fax machine). AFAX = sent via Fax machine.

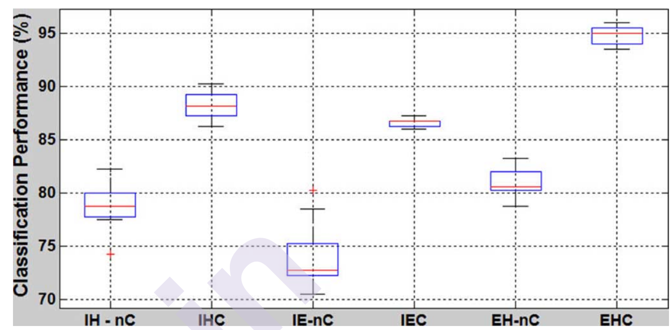



Fig. 12. Box plot of degradation classification performance results when using a combination of features. Note that the central mark (red line) is the median classification result over 10 runs, the edges of the box (blue) are the 25th and 75th percentiles, the whiskers (black lines) extend to the most extreme data points not considered outliers, and outliers (red crosses) are plotted individually. $I$ = Inertia; $H$ = Homogeneity; $E$ = Energy; $C$ = Contrast (Image Graininess); and $nC$ = no usage of Contrast.

option, we achieve an average quality improvement of about 54% (PSNR), or approximately 7 times in terms of UIQ.

We note that the PSNR method does not provide as crisp a result as UIQ leading us to the following question: which metric should be trusted? We know that PSNR (as well as mean squared error) is one of the most widely used objective image quality/distortion metric, but is widely criticized as well, for not correlating well with perceived quality measurement. There are many other image quality measurements proposed in the literature, but most of them share a common error-sensitivity-based philosophy (motivated from psychological vision science research), i.e., human visual error sensitivities and masking effects vary in different spatial frequency, temporal frequency, and directional channels.

In our experiments, UIQ appears to be more robust in the selection of the best reconstructed image. Even though both PSNR and UIQ lead to the same conclusion (that the combination of image filtering and TI denoising is preferable), they converge to a different filtering window size and level of thresholding, and ultimately image restoration quality. Fig. 10 illustrates some cases where the reconstructed images based on PSNR were not as good as those that were based on UIQ. This is a general conclusion based on the results found across all degradation scenarios investigated in this paper.

Based on the results obtained in this set of experiments, we applied the iterative TI-wavelet restoration algorithm that combines Hanning filtering and Symlets with hard thresholding to both passport and passport-fax databases. The quality of the restoration was then tested by using the commercial face recognition software provided by L1 Systems.[4]

### B. Evaluation of the Degradation Classification Algorithm

The *second* experimental scenario illustrates the efficiency of the degradation classification algorithm, i.e., the capability of identifying the degradation type of an input image. For each degraded dataset generated from the *FRGC2-Passport FAX Database* (Section V), a subset (approximately 22.5% of the training set that was used for the identification experiments) is used to extract the textural features as well as image graininess. Out of all the features considered here, the optimal ones in terms of performance are energy, homogeneity, and graininess. In Fig. 11(a), we see the clustering of these feature sets based on the nature of degradation of the input image. It is important to see that images in datasets 1 and 2 are within the same cluster. In contrast, datasets 3 and 4 form their own clusters. In addition, image graininess can be used to separate datasets (1,2) from (3,4).

[4]Available: http://www.l1id.com/

TABLE I
CLASSIFICATION RESULTS WHEN USING THE TEXTURAL- AND QUALITY-BASED CLASSIFICATION ALGORITHM. B4FAX = FAX COMPRESSION (NOT SENT VIA FAX MACHINE); LRes = LOW RESOLUTION; HRes = HIGH RESOLUTION; AFAX = SENT VIA FAX MACHINE; CL = CLASSIFICATION; EV = ERROR VARIANCE

| FEATURES | DATASETS | CL (%) | EV |
|---|---|---|---|
| TEXTURAL | Class 1: [BFAX LRes/HRes]; Class 2: BFAX (Print and Scan); Class 3: AFAX | 89.34 | 0.039 |
| GRAININESS | Class 1: [BFAX LRes/HRes]; Class 2: BFAX (Print and Scan) and AFAX data | 100.00 | - |
| FUSED | Class 1: [BFAX LRes/HRes]; Class 2: BFAX (Print and Scan); Class 3: AFAX | 96.84 | 0.038 |

TABLE II
PARTITIONING THE *FRGC2-PASSPORT* FAX DATABASE PRIOR TO APPLYING THE CSU FR ALGORITHMS

| | TRAIN | TEST | GALLERY | PROBES |
|---|---|---|---|---|
| Subjects # | 125 | 283 | 283 | 283 |
| Images/Subject | 4 | 4 | 0.25 (from TEST) | 2-4 (from TEST) |

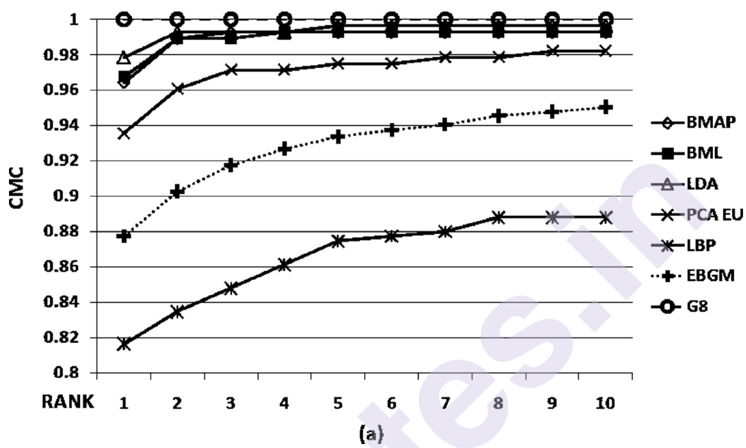



Fig. 13. Face identification results: High-quality versus high-quality face image comparison.

To test our classification algorithm, we used a dataset of $N = 108$ sample images (27 subjects $\times$ 4) for training and the four samples of the remaining (28th) subject for testing (1 subject $\times$ 4), where 4 in both cases represents one sample from each of the four classes involved. Thus, we performed a total of 28 experiments where the training and test datasets were resampled, i.e., in each experiment the data of a different subject (out of the 28) was used for testing. Each experiment was performed before and after fusing textural and image graininess features. The results are summarized in Table I. Note that if an image is misclassified, it will be subjected to the set of meta-parameters pertaining to the incorrect class.

We also applied our feature extraction algorithm on the original training set of the FRGC2 subset. Then, we randomly selected 100 samples from the original test set (see Table II) 10 times, and then applied feature extraction on each generated test subset. We performed the above process on the three degraded datasets that are generated from the original FRGC2 training/test sets, and performed 26 400 classification experiments in total. The outcome of these experiments is summarized Fig. 12 (box-plot results).

## C. *Face Identification Experiments*

The *third* experimental scenario is a series of face identification tests which compare system performance resulting from the baseline (FRGC2-Passport FAX Database), degraded, and reconstructed face datasets. The goal here is to illustrate that the face matching performance improves with image restoration. For this purpose, we perform a two-stage investigation that involves 1) high-quality versus high-quality face image comparison (baseline), and 2) high-quality versus degraded face image comparison. In the *high-quality versus high-quality* tests, we seek to establish the baseline performance of each of the face recognition methods (academic and commercial) employed. In the *high-quality versus degraded* tests, we investigate the matching performance of degraded face images against their high-resolution counterparts.

Table II illustrates the way we split the *FRGC2-Passport FAX Database* to apply the CSU FR algorithms. For the G8 and LBP algorithms we used 4 samples of all the 408 subjects, and ran a 5-fold cross-validation where one sample per subject was used as the gallery image and the rest were used as probes. The *identification* performance of the system is evaluated through the cumulative match characteristic (CMC) curve. The CMC curve measures the $1 : m$ identification system performance, and judges the ranking capability of the identification system.

All the results before and after restoration are presented in Figs. 13–16. We can now evaluate the consistency of the results and the significant benefits of our restoration methodology in terms of face identification performance. For high-quality face images with no photometric normalization, the average *rank 1 score* of all the FR algorithms is ~93.43% (see Fig. 13). This average performance drops to 80.4% when fax compression images are used before restoration. After restoration, the average *rank 1 score* increases to 90.8% (12.94% performance improvement). When the fax compressed images are also printed,

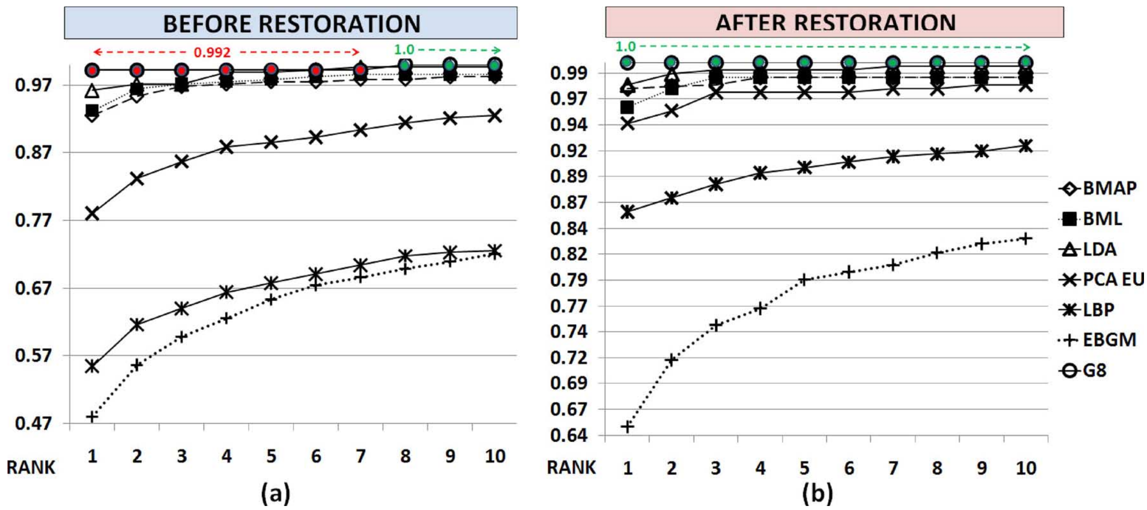



Fig. 14. Face identification results: High-quality versus Fax compressed images.

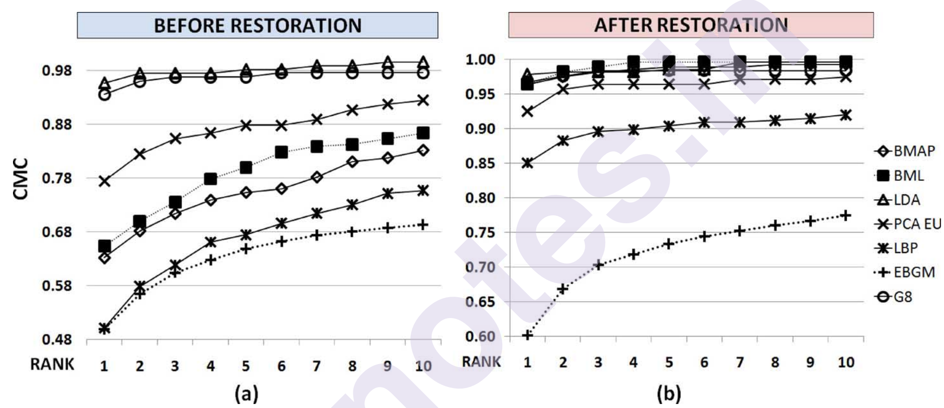



Fig. 15. Face identification results: High-quality versus Fax compressed images which have been printed and scanned.

the performance drops further to 70.8% before restoration, but increases to 89.3% after restoration (26.13% performance improvement). Finally, when the most degraded images were used (images sent via a fax machine) the average *rank 1 score* across all the algorithms drops to 58.7% before restoration while after restoration it goes up to 81.2%. It is interesting to note that the identification performance of the high-quality images is comparable to that of the restored degraded images. Note that each face identification algorithm performs differently, and in some cases (e.g., G8), the performance is optimal for both raw and restored images (in the case of fax compression) achieving a 100% identification rate at rank 1. The consistency in improving recognition performance indicates the significance of the proposed face image restoration methodology.

## VII. Conclusions and Future Work

We have studied the problem of image restoration of severely degraded face images. The proposed image restoration algorithm compensates for some of the common degradations encountered in a law-enforcement scenario. The proposed restoration method consists of an offline mode (image restoration is applied iteratively, resulting in the optimum meta-parameter sets), where the objective function is based on two different image quality metrics. The online restoration mode uses a classification algorithm to determine the nature of the degradation in the input image, and then uses the meta-parameter set identified in the offline mode to restore the degraded image. Experimental results show that the restored face images not only have higher image quality, but they also lead to higher recognition performance than their original degraded counterparts.

Commercial face recognition software may have their own internal normalization schemes (geometric and photometric) that cannot be controlled by the end-user, and this can result in inferior performance when compared to some academic algorithms (i.e., LDA) when restoration is employed. For example, when G8 was used on fax compressed data, the identification performance was 79.2% while LDA resulted in a 91.4% matching accuracy. In both cases, the restoration helped, yet LDA (97.9%) performed better than G8 (93.6%). Since the preprocessing stage of the noncommercial algorithms can be better controlled than commercial ones, several academic algorithms were found to be comparable in performance to the commercial one after restoration.

The proposed image restoration approach can potentially discard important textural information from the face image. One possible improvement could be the use of super-resolution algorithms that learn *a prior* on the spatial distribution of the image

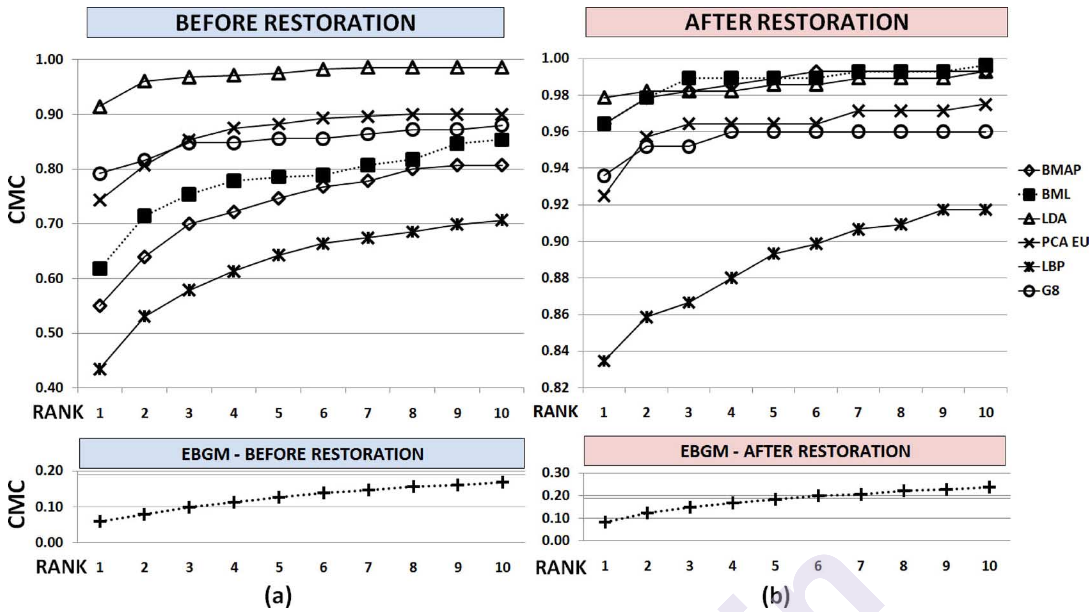



Fig. 16. Face identification results: High-quality versus images that are sent via a Fax machine and then scanned. Note that the EBGM method is illustrated separately because it results in very poor matching performance. This could be implementation-specific and may be due to errors in detecting landmark points.

gradient for frontal images of faces [19]. Another future direction is to extend the proposed approach to real surveillance scenarios in order to restore low quality images. Finally, another area that merits further investigation is the better classification of degraded images. Such an effort will improve the integrity of the overall restoration approach.

## ACKNOWLEDGMENT

The authors would like to thank researchers at Colorado State University for their excellent support in using the Face Evaluation Toolkit. They are grateful to Z. Jafri, C. Whitelam, and A. Jagannathan at West Virginia University for their valuable assistance with the experiments.

## REFERENCES

[1] T. Bourlai, A. Ross, and A. Jain, "On matching digital face images against scanned passport photos," in *Proc. First IEEE Int. Conf. Biometrics, Identity and Security (BIDS)*, Tampa, FL, Sep. 2009.
[2] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe, "FRVT 2006 and ICE 2006 large-scale experimental results," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 831–846, May 2010.
[3] S. K. Zhou, R. Chellappa, and W. Zhao*, Unconstrained Face Recognition*. New York: Springer, 2006.
[4] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *Proc. Society of Information Display*, 1976, vol. 17, pp. 75–77.
[5] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
[6] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Proc. Computer Vision and Pattern Recognition Conf.*, Jun. 2005, vol. 1, pp. 947–954.
[7] T. Ahonen, A. Hadid, and M. Pietikinen, "Face recognition with local binary patterns: Application to face recognition," in *Proc. Eur. Conf. Computer Vision (ECCV)*, Jun. 2004, vol. 8, pp. 469–481.
[8] D. S. Bolme, J. R. Beveridge, M. L. Teixeira, and B. A. Draper, "The CSU face identification evaluation system: Its purpose, features and structure," in *Proc. Int. Conf. Computer Vision Systems*, Apr. 2003, pp. 304–311.
[9] H. C. Andrews and B. R. Hunt*, Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
[10] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Process. Mag.*, vol. 14, no. 2, pp. 24–41, Aug. 2002 [Online]. Available: http://dx.doi.org/10.1109/79.581363
[11] J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially-variant blur," *SIAM J. Sci. Comput.*, vol. 19, pp. 1063–1082, 1996.
[12] M. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
[13] J. Bioucas Dias and M. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.
[14] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, "A study of methods of choosing the smoothing parameter in image restoration by regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 326–339, Apr. 1991.
[15] M. I. Sezan and A. M. Tekalp, "Survey of recent developments in digital image restoration," *Opt. Eng.*, vol. 29, no. 5, pp. 393–404, 1990 [Online]. Available: http://link.aip.org/link/?JOE/29/393/1
[16] P. H. Hennings-Yeomans, S. Baker, and B. V. Kumar, "Simultaneous super-resolution and feature extraction for recognition of low-resolution faces," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2008, pp. 1–8.
[17] P. H. Hennings-Yeomans, B. V. K. V. Kumar, and S. Baker, "Robust low-resolution face identification and verification using high-resolution features," in *Proc. Int. Conf. Image Processing (ICIP)*, Nov. 2009, pp. 33–36.
[18] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example based super-resolution," *IEEE Comput. Graph. Applicat.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
[19] S. Baker and T. Kanade, "Hallucinating faces," in *Proc. Fourth Int. Conf. Auth. Face and Gesture Rec.*, Grenoble, France, 2000.
[20] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 817–834, Sep. 1999.
[21] N. Ramanathan and R. Chellappa, "Face verification across age progression," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3349–3362, Nov. 2006.
[22] V. V. Starovoitov, D. Samal, and B. Sankur, "Matching of faces in camera images and document photographs," in *Proc. Int. Conf. Acoustic, Speech, and Signal Processing*, Jun. 2000, vol. IV, pp. 2349–2352.

[23] V. V. Starovoitov, D. I. Samal, and D. V. Briliuk, "Three approaches for face recognition," in *Proc. Int. Conf. Pattern Recognition and Image Analysis*, Oct. 2002, pp. 707–711.

[24] S. K. Mohideen, S. A. Perumal, and M. M. Sathik, "Image de-noising using discrete wavelet transform," *Int. J. Comput. Sci. Netw. Security*, vol. 8, no. 1, pp. 213–216, Jan. 2008.

[25] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.

[26] D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[27] R. R. Coifman and D. L. Donoho, "Translation-invariant de-noising," in *Wavelets and Statistics*. New York: Springer-Verlag, 1994, vol. 103, Springer Lecture Notes, pp. 125–150.

[28] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[29] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[30] E. Fix and J. L. Hodges, Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties USAF School of Aviation Medicine, Randolph Field, TX, Tech. Rep. 4, 1951.

[31] Setup and Production Guidelines for Passport and Visa Photographs U.S. Department of State, 2009 [Online]. Available: http://travel.state.gov/passport/get/get_873.html

[32] P. A. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[33] L. Sirovich and M. Kirby, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

[34] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[35] A. P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[36] P. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[37] M. Teixeira, "The Bayesian Intrapersonal/Extrapersonal Classifier," Master's thesis, Colorado State University, Fort Collins, CO, 2003.

[38] L. Wiskott, J.-M. Fellous, N. Kruger, and C. V. D. Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, Jul. 1997.

[39] M. Pietikinen, "Image analysis with local binary patterns," in *Proc. Scandinavian Conf. Image Analysis*, Jun. 2005, pp. 115–118.


**Thirimachos Bourlai** (M'10) received the Diploma (M.Eng. equivalent) in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1999, the M.Sc. degree in medical imaging (with distinction) from the University of Surrey, U.K., in 2002 under the supervision of Prof. M. Petrou. He received the Ph.D. degree (full scholarship) in the field of face recognition and smart cards, in 2006, in a collaboration with OmniPerception Ltd. (U.K.), and his Postdocorate in multimodal biometrics, in August 2007, both under the supervision of Prof. J. Kittler.

He worked as a Postdoctoral researcher in a joint project between the University of Houston and the Methodist Hospital (Department of Surgery) at Houston, TX, in the fields of thermal imaging and computational physiology. From February 2008 to December 2009 he worked as a Visiting Research Assistant Professor at West Virginia University (WVU), Morgantown. Since January 2010 he has been a Research Assistant Professor at WVU. He is supervising the eye detection team, has been involved in various projects in the fields of biometrics, and multispectral imaging, and authored several book chapters, journals and conference papers. His areas of expertise are image processing, pattern recognition, and biometrics.

**Arun Ross** (S'00–M'03–SM'10) received the B.E. (Hons.) degree in computer science from BITS, Pilani, India, in 1996, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, in 1999 and 2003, respectively.

Between 1996 and 1997, he was with Tata Elxsi (India) Ltd., Bangalore. He also spent three summers (2000–2002) at Siemens Corporate Research, Inc., Princeton working on fingerprint recognition algorithms. He is currently an Associate Professor in the Lane Department of Computer Science and Electrical Engineering at West Virginia University. His research interests include pattern recognition, classifier fusion, machine learning, computer vision, and biometrics. He is the coauthor of *Handbook of Multibiometrics* and coeditor of *Handbook of Biometrics*. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

Dr. Ross is a recipient of NSF's CAREER Award and was designated a Kavli Frontier Fellow by the National Academy of Sciences in 2006.

**Anil K. Jain** (S'70–M'72–SM'86–F'91) is a university distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. The holder of six patents in the area of fingerprints, he is the author of a number of books, including *Handbook of Fingerprint Recognition* (2009), *Handbook of Biometrics* (2007), *Handbook of Multibiometrics* (2006), *Handbook of Face Recognition* (2005), *BIOMETRICS: Personal Identification in Networked Society* (1999), and *Algorithms for Clustering Data* (1988).

Dr. Jain received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award and the Pattern Recognition Society best paper awards in 1987, 1991, and 2005. He served as the editor-in-chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (1991–1994). He is a fellow of the AAAS, ACM, IEEE, IAPR, and SPIE. He has received Fulbright, Guggenheim, Alexander von Humboldt, IEEE Computer Society Technical Achievement, IEEE Wallace McDowell, ICDM Research Contributions, and IAPR King-Sun Fu awards. ISI has designated him a highly cited researcher. According to Citeseer, his book *Algorithms for Clustering Data* (Prentice-Hall, 1988) is ranked #93 in most cited articles in computer science. He served as a member of the Defense Science Board and The National Academies committees on Whither Biometrics and Improvised Explosive Devices.