Image Processing

Ghufran Saady

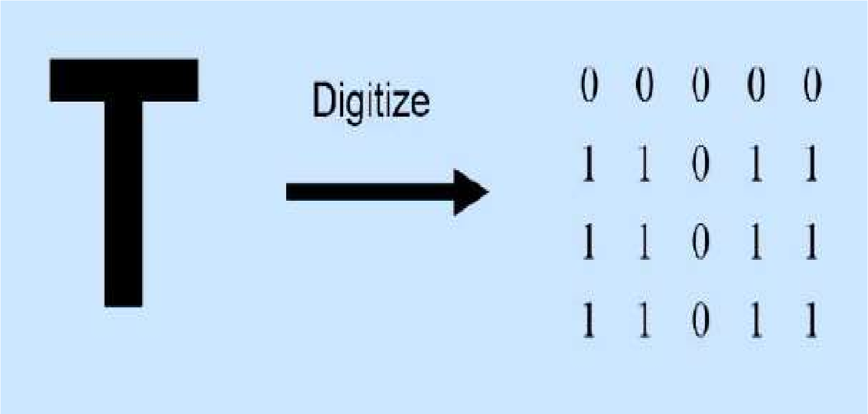
Image Processing

- Practically every scene around us involves images or image processing.
- An image can be defined as a two dimensional signal analog or
- digital that contains intensity or color information arranged along x and y spatial axis.
- It can be defined as a two dimensional function f(x,y) where the x and y are spatial co-ordinates. Here the amplitude of function “ f ” at any pair of co-ordinates (x,y) is called theintensity or gray level or the color of the image at that point.

Image Types

□ There are two types of images, **analog** and **digital.**

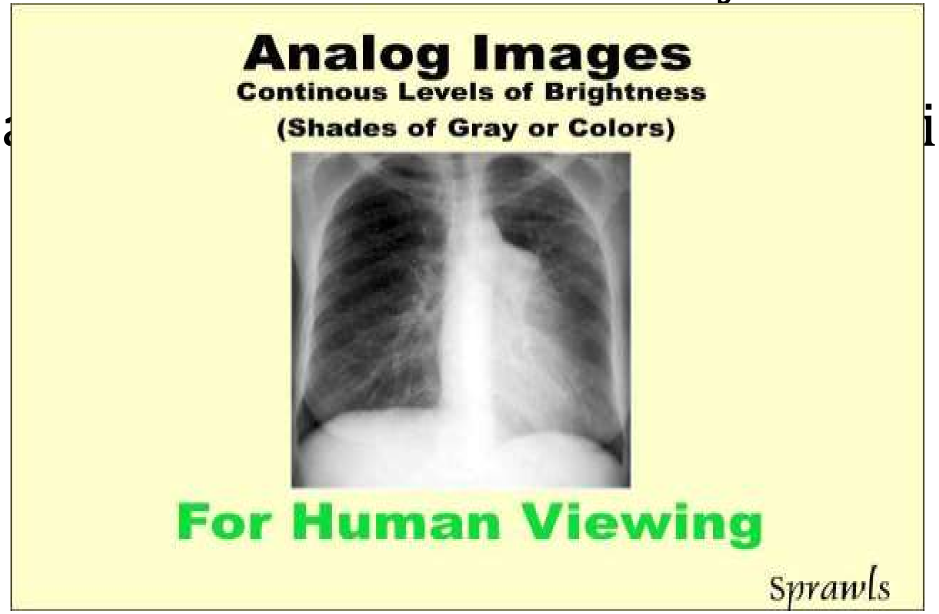
□ Simple example, let 1 represent white and 0 represent black
color.



- **Analog Image**
- Analog image is a two dimensional function of f(s,t) considered in the continuous time domain.
- An image can be defined as a two dimensional function f(x,y) where the x and y are spatial co-ordinates. Here the amplitude of function " f " at any pair of co-ordinates (x,y) is called the intensity or gray level or the color of the image at that point.
- When x, y and the amplitude values of f are continuous quantities then the image is referred as analog image.

Analog Image

- Analog images are the type of images that we, as humans, look at. They also include such things as photographs, paintings, TV images, and all of our medical images recorded on film or displayed on various display devices.
- What we see in an analog image is various levels of brightness (or film density) and colors. It is generally continuous and not broken into many small individual pieces.
- Analog ima … ing.



- **Digital Image**
- An image can be defined as a two dimensional function f(x,y) where the x and y are spatial co-ordinates. Here the amplitude of function “ f ” at any pair of co-ordinates (x,y) is called the intensity or gray level or the color of the image at that point.
- When x, y and the amplitude values of f are all finite and discrete quantities then the image is referred as digital image. Digital image processing refers to the processing of digital images by means of digital computer.
- A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as pels, pixels, picture elements or image elements.

Digital Image

 A digital image is a matrix of many small elements, or pixels.

 Each pixel is represented by a numerical value. In general, the pixel value is related to the brightness or color that we will see when the digital image is converted into an analog image for display and viewing.

Impact Digital Image

 Digital images are necessary in all modern medical imaging methods. Because of the following functions that can be performed with digital images :

1. Image reconstruction (CT, MRI, SPECT, PET, etc)

2. Image reformatting (Multi-plane, multi-view reconstructions)

3. Wide (dynamic) range image data acquisition (CT, digital radiography, etc)

4. Image processing (to change contrast and other quality characteristics)

5. Fast image storage and retrieval

6. Fast and high-quality image distribution (teleradiology)

7. Controlled viewing (windowing, zooming, etc)

8. Image analysis (measurements, calculation of various parameters, computer aided diagonisis

Analog VS Digital Image

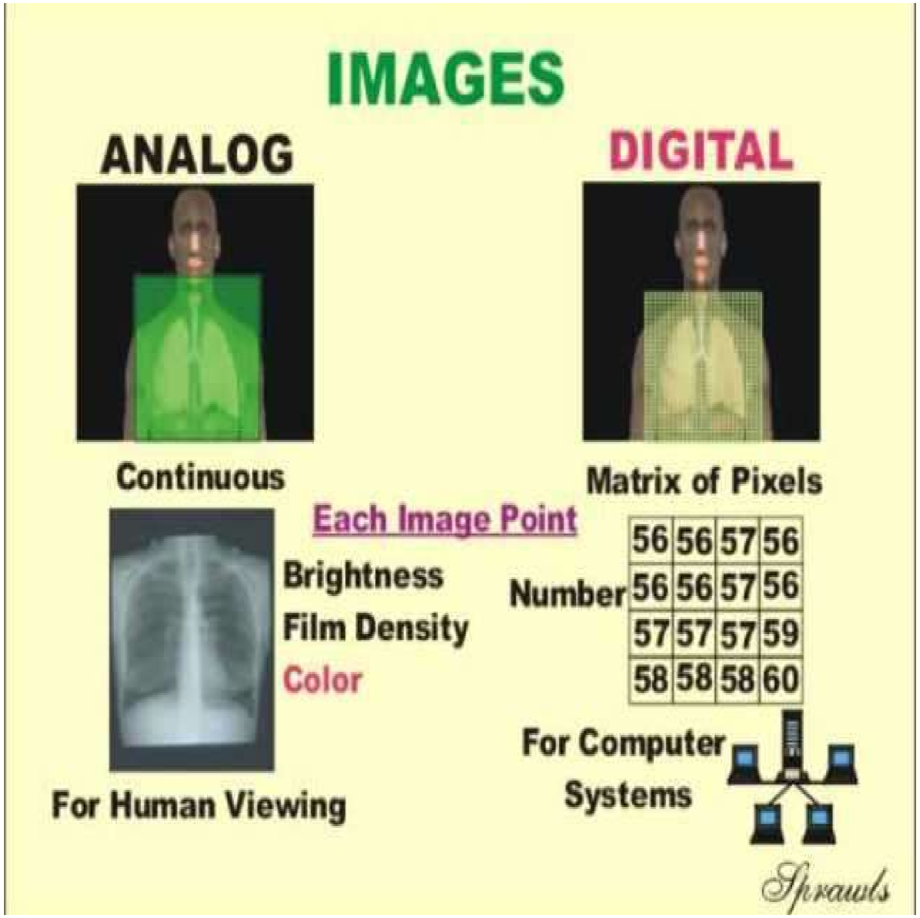


Image Processing

 Some people consider that image processing is a discipline in which both the input and output of a process are images. According to this definition computing average intensity of an image would not be considered as image processing task.

However there is no such limitations or boundaries for image
processing.

 However, one useful paradigm is to consider three types of computerized processes in this continuum:

1. low-level image processing
2. mid-level image processing and
3. high-level image processing

Level of Image Processing

- A low-level process is characterized by the fact that both its inputs and outputs are images such as image preprocessing to reduce noise, contrast enhancement, and image sharpening.
- A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images. Such as segmentation (partitioning an image into regions or objects).
- A high-level process is characterized by the fact that

Level of Image Processing

 A low-level process is characterized by the fact that both its inputs and outputs are images such as image preprocessing to reduce noise, contrast enhancement, and image sharpening.

 A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images. Such as segmentation (partitioning an image into regions or objects).

 A high-level process is characterized by the fact that its inputs generally are attributes extracted from images, but

Origins of Digital Image Processing

 One of the first applications of digital images was in the newspaper industry, when pictures were first sent by submarine cable between London and New York.

 Introduction of the Bartlane cable picture transmission system in
early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours.

 Specialized printing equipment was used to code

Digital Image Processing

 Digital image processing can be defined as processing of digital image in a digital manner meaning that using a digital device like computer or others.

 The digital image processing is getting more and more importance now a days because of its two major application areas:

1. Improvement of pictorial information for human interpretation.

2. Processing of image data for storage, transmission and

Advantages of DIP

1. It improves the visual quality of an image and the distribution of intensity.

2. It can easily process an degraded image of uncoverable objects

3. It can process an image in such a way that the result is more suitable than the original image

4. An image can be easily modified using a number of techniques

5. The image compression technique reduces the amount of data required to represent a digital image.

6. Mathematical and logical operations can be performed on

Limitations of DIP

1. Digital image processing requires so much storage and processing power. Progress in the field of digital image processing is dependant on the development of digital computers and supporting technology including data storage, display and transmission

2. Effect of environmental conditions may degrade the image quality

3. It involves various types of redundancy like data redundancy, interpixel redundancy etc

4. Segmentation of nontrivial image is one of the most difficult task in digital image processing

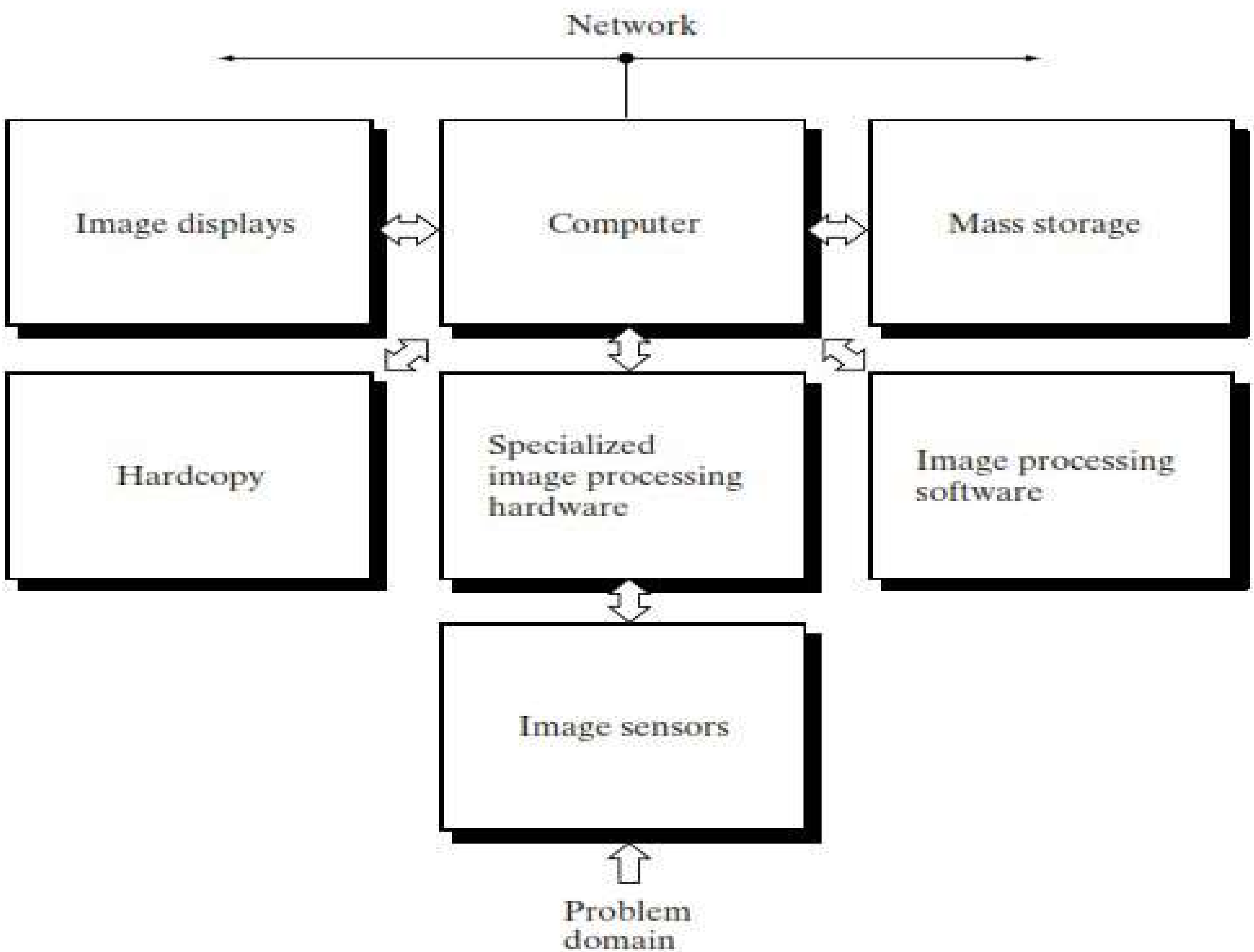
Fields that Use Digital Image Processing

 Unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that humans are not accustomed to associating with images.

1. Gamma ray imaging
2. X-ray imaging
3. Imaging in an ultraviolet band
4. Imaging in the visible and infrared bands
5. Imaging in the microwave band

Components of Image Processing System

 Following figure shows the basic components comprising a typical *general-purpose* system used for digital image processing



Components of the System

- **Image Sensor:** With reference to sensing, two elements are required to acquire digital images: **a sensor and a digitizer**. The sensor that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For example, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

- **Specialized image processing hardware:** usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU). ALU performs arithmetic and logical operations in parallel on entire images. ALU is used is in averaging images as quickly as they are digitized, for the purpose of noise reduction. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed in

Components of the System

- **Computer:** in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, some times specially designed computers are used to achieve a required level of performance.
- **Software:** for image processing consists of specialized modules that perform specific tasks
- **Mass storage:** capability is a must in image processing applications. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access.
- **Image display:** it displays images.
- **Hardcopy devices:** used for recording images include laser printers, film cameras, heat sensitive devices, inkjet units,

Types of Images

Types of images

There are many type of images , and we will look in detail about different types of images….

- The binary image
- Gray level images
- Color images
- Multispectral images

Binary images:

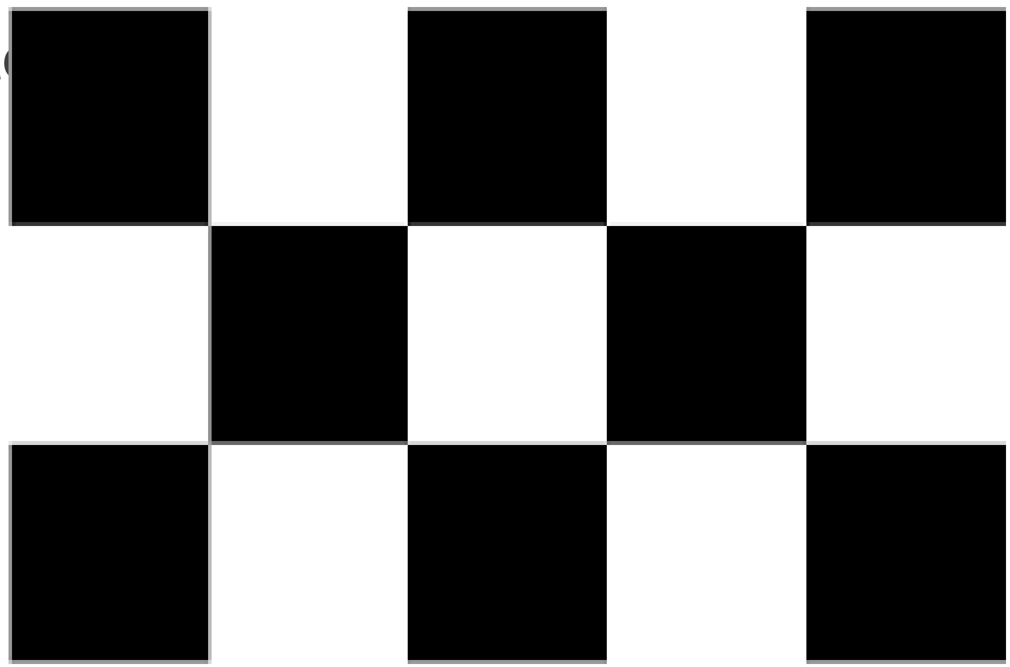
- The binary image as it name states , contain only two pixel values. 0 and 1.

Binary images also known as black & white images.....

Here 0 refers to black color and 1 refers to white color. It is also known as Monochrome.

Black and white images

- Images stored as two dimensional (m-by-n)array of logical(0 or 1)
- The resulting image that is formed hence consist of only black and white color and thus can also be called as Black and Whit



NOT INCLUDE GRAY LEVEL

One of the interesting this about this binary image that there is no gray level in it. Only two colors that are black and white are found in it.

Gray scale images:

(8 bits per pixel)

- Images is store as a two dimensional (m-by-n)array of integers in the range

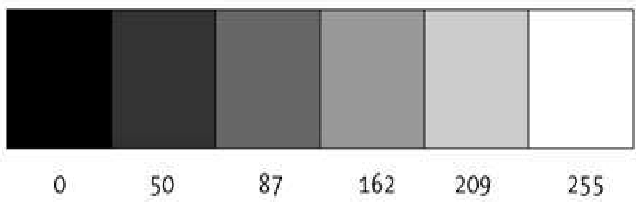
0-255

It is often called intensity images

In photography and computing, a **grayscale** or **greyscale** digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information.

Images of this sort, also known as black-and-white, are composed exclusively of

Representation of gray scale images:



Colors images

- The human visual system can distinguish hundreds of thousands of different color shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the color, and this extra information can then be used to simplify image analysis, e.g. object identification and extra

Color images:

There are two types of images

- True color(RGB)
- Indexed(colormap)

True Color images

- Truecolor is a means to specify a color explicitly with RGB values rather than pointing to an entry in the figure colormap. Truecolor generally provides a greater range of colors than can be defined in a colormap.

Image is stored as a three-dimensional (m-by-n-by-3) array of integers i

Indexed color

- In digital photography and imaging, indexed color is the term used to describe reduced color mapping of 8-bit or less. This is done to reduce images to their smallest size and these images are most commonly used on Web pages as they are small and quick to load. The 256 color palette is mapped for best results on the Internet, taking into account the differences between the Windows and Macintosh color palettes.
- Image is stored as a two-dimensional (m-by-n-by-2)

DIGITAL IMAGE PROCESSINGIMAGE ENHANCEMENT Neighborhood Pixels Processing

Neighborhood Pixels Processing

- It is also spatial domain technique in image enhancement.
- Here, we consider one pixel at a time & modify it accordingly.
- Its neighboring pixels are also taken in consideration.
- So, we change pixel value based on 8 neighbors.
- Along with 3x3 neighborhood, 5x5 & 7x7 can also be used.
- A lot of things can be achieved by neighborhood processing not possible by point processing.

Once f(x, y) is calculated, shift mask by 1 step to right.

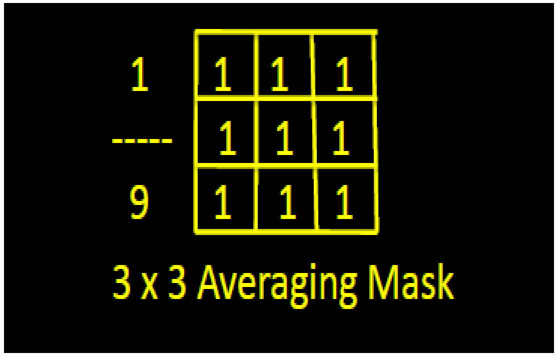
- Application of neighborhood processing : Image Filtering.
- E.g. LPF, HPF, BPF, BRF
- In 1D signals, if 2 signals represent voltage then,
- How fast the signal changes is indication of frequency.
- Same concept is applied to images where we have gray levels instead.
- If gray scale change slowly over a region then LF area. E.g. Background
- If gray scale change abruptly over a region then HF area. E.g. Edges, Boundaries.

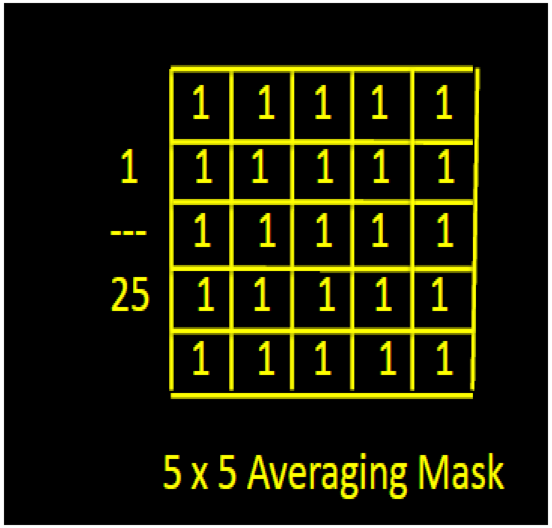
Low Pass Filtering (Smoothing):

- Removes HF content from image.
- Used to remove noise (HF component) from image.
- Noise: Noise creeps in during image acquisition & transmission. Noises are classified as:
- i) Gaussian Noise
- ii) Salt & Pepper Noise
- iii) Rayleigh Noise
- iv) Gamma Noise
- v) Exponential Noise
- vi) Uniform Noise

Low Pass Averaging filter:

- Generally used for removal of Gaussian noise from images.
- It uses a mask that gives LPF operation.
- Important thing: All the coefficients are positive.
- Standard LPF Averaging masks:





Ex. 1) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size averaging mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

Neighborhood Pixels Processing

0	0	0						
0	10	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

0	0	0						
0	4.44	10	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 10 \\ 0 & 10 & 10 \end{bmatrix}$$

0	4.44	6.66	10	10	10	10	10	10
0	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50
	50	50	50	50	50	50	50	50

1	0	0	0
-----	10	10	10

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	10	0
	10	10	10	10	10	10	10	10	
	10	10	10	10	10	10	10	10	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	
	50	50	50	50	50	50	50	50	

1	10	10	10
-----	10	10	10
9	10	10	10

0	0	0	0	0	0	0	0	0	
0	4.44	6.66	6.66	6.66	6.66	6.66	6.66	4.44	0
0	6.66	10	10	10	10	10	10	6.66	0
0	6.66	10	10	10	10	10	10	6.66	0
0	15.55	23.33	23.33	23.33	23.33	23.33	23.33	15.55	0
0	24.44	36.66	36.66	36.66	36.66	36.66	36.66	24.44	0
0	33.33	50	50	50	50	50	50	33.33	0
0	33.33	50	50	50	50	50	50	33.33	0
0	22.22	33.33	33.33	33.33	33.33	33.33	33.33	22.22	0
	0	0	0	0	0	0	0	0	0

1

9

50	50	0
50	50	0
0	0	0

- In the resultant image the Low frequency region has remained unchanged.
- Sharp transition between 10 & 50 has changed from 10 to 23.33 to 36.66 and finally to 50.
- Thus, Sharp edges has become blurred.
- Best result when used over image corrupted by Gaussian noise.

- Other types of low pass averaging mask are:

$$\frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Median Filtering:

- Averaging Filter removes the noise by blurring till it is no longer seen.
- It blurs the edges too.
- Bigger the averaging mass more the blurring.
- Sometimes the image contains 'salt & pepper noise'.
- If averaging filter is used then it will remove the noise at the cost of ruined edges.
- Thus a nonlinear filter Median filter is required.
- They are also called as order statistics filter since their response is based on ordering or ranking of pixels contained within the mask.

Steps to perform median filtering:

- Assume a 3x3 empty mask.
- Place the empty mask at the Left Hand corner.
- Arrange the 9 pixels in ascending or descending order.
- Choose the median from these 9 values.
- Place the median at the centre.
- Move the mask in same manner as averaging filter

- Apply 3x3 median filter to find a new image.

3	4	2	3
1	7	3	2
4	5	3	8
2	3	1	7

Noisy Image S & P noise

3x3 blank mask

❑ Apply 3x3 median filter to find a new image.

3	4	2	3
1	7	3	2
4	5	3	8
2	3	1	7

3	4	2	3
1	3		2
4			8
2	3	1	7

1) 1 2 3 3 3 4 4 5 7

- Apply 3x3 median filter to find a new image.

3	4	2	3
1	7	3	2
4	5	3	8
2	3	1	7

3	4	2	3
1	3	3	2
4			8
2	3	1	7

1) 1 2 3 3 3 4 4 5 7

2) 2 2 3 3 3 4 5 7 8

3	4	2	3
1	7	3	2
4	5	3	8
2	3	1	7

3	4	2	3
1	3	3	2
4	3		8
2	3	1	7

1) 1 2 3 3 3 4 4 5 7

2) 2 2 3 3 3 4 5 7 8

3) 1 1 2 3 3 3 4 5 7

3	4	2	3
1	7	3	2
4	5	3	8
2	3	1	7

3	4	2	3
1	3	3	2
4	3	3	8
2	3	1	7

1) 1 2 3 3 3 4 4 5 7
2) 2 2 3 3 3 4 5 7 8
3) 1 1 2 3 3 3 4 5 7
4) 1 2 3 3 3 5 7 7 8

Ex. 2) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 50. Remove using a 3x3 size median filter mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	250	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

8x8 Image with blank mask

Ex 4). Find the median filtered image by 3x3 mask for the given image.

2	4	15	0
3	5	2	6
11	0	2	10
6	16	0	2

High Pass Filtering:

- Retains HF component while eliminates LF components.
- High passed image will have no background(Low freq region).
- It will have enhanced edges.
- Used to sharpen blurred images.
- Process of mask moving on image is same only the mask coefficients change.
- Mask coefficients should have positive value at centre and negative values elsewhere.
- Sum of coefficients must be zero.
- Since, it should give Zero after being placed on LP region.

High Pass Masks:

3x3 High pass masks

-1	-1	-1
-1	8	-1
-1	-1	-1

0	-1	0
-1	4	-1
0	-1	0

-1	-2	-1
-2	12	-2
-1	-2	-1

Ex 5) 8x8 Pseudo image with a single edge (High Frequency) of 10 & 100. Remove LP using a 3x3 size High pass filter mask.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

-1	-1	-1
-1	8	-1
-1	-1	-1

-10-10-10-10-10-10-10-10+80 = 0

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

-1	-1	-1
-1	8	-1
-1	-1	-1

$$-10-10-10-10-10-10-10-10+80 = 0$$

Neighborhood Pixels Processing

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

-1	-1	-1
-1	8	-1
-1	-1	-1

$-10-10-10-10-10-100-100-100+80 = -270$

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

-1	-1	-1
-1	8	-1
-1	-1	-1

$-10-10-10-100-100-100-100-100+800 = +270$

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100

-1	-1	-1
-1	8	-1
-1	-1	-1

$-100-100-100-100-100-100-100-100+800 = 0$

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-270	-270	-270	-270	-270	-270	-270	-270
270	270	270	270	270	270	270	270
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Note: -270 is replaced by 0.

DIGITAL IMAGE PROCESSING

IMAGE COMPRESSION

Introduction

- Image Compression: It is the Art & Science of reducing the amount of data required to represent an image.

- It is the most useful and commercially successful technologies in the field of Digital Image Processing.

- The number of images compressed and decompressed daily is innumerable.

Fundamentals

- Data Compression: It refers to the process of reducing the amount of data required to represent a given quantity of information.

Data

Vs

Information

- Data and Information are not the same thing; data are the means by which information is conveyed.

- Because various amount of data can be used to represent the same amount of information, representations that contain irrelevant or repeated information are said to contain redundant data.

Fundamentals

- 2D intensity arrays suffers from 3 principal types of data redundancies:

1) Coding redundancy: A code is a system of symbols used to represent a body of information or sets of events.

- Each piece of event is assigned a code word (code symbol). The number of symbols in each code word is its length.

- The 8-bit codes that are used to represent the intensities in most 2D intensity arrays contain more bits than are needed to represent the intensities.

2) Spatial & Temporal redundancy:

- Because the pixels of most 2D intensity arrays are correlated spatially, information is replicated unnecessarily.
- In video sequence, temporally correlated pixels also duplicate information.

3) Irrelevant Information:

- Most 2D intensity arrays contain information that is ignored by the human visual system. It is redundant in the sense that it is not used.

Fundamentals

- Let b & b’ denote the number of bits in two representations of the same information, the relative data redundancy R of the representation with b bits is

- $R = 1 – (1/C)$;

  where, C commonly called the compression ratio, is defined as
- $C = b / b'$

- If C = 10 (or 10:1), for larger representation has 10 bits of data for every 1 bit of data in smaller representation.

So, R = 0.9, indicating that 90% of its data is redundant.

Image Compression Models

- The image compression system is composed of 2 distinct functional component: an encoder & a decoder.
- Encoder performs Compression

  while
- Decoder performs Decompression.
- Both operations can be performed in Software, as in case of Web browsers & many commercial image editing programs.
- Or in a combination of hardware & firmware, as in DVD Players.
- A codec is a device which performs coding & decoding.

Image Compression Models

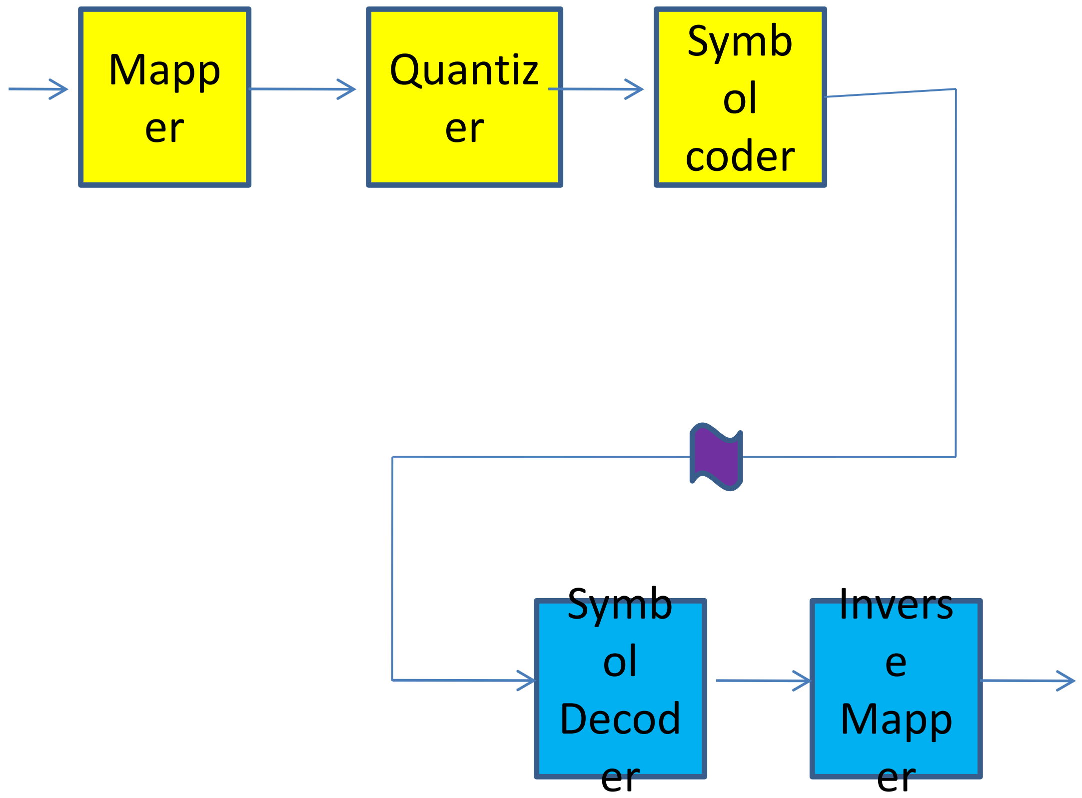


Image Compression Models

Encoding or Compression process:

Encoder is used to remove the redundancies through a series of 3 independent operations.

Mapper: It transforms f(x,…) into a format designed to reduce spatial and temporal redundancies.

- It is reversible
- It may / may not reduce the amount of data to represent image.

Ex. Run Length coding

- In video applications, mapper uses previous frames to remove temporal redundancies.

Image Compression Models

Quantizer: It keeps irrelevant information out of compressed representations.

- This operation is irreversible.
- It must be omitted when error free compression is desired.
- In video applications, bit rate of encoded output is often measured and used to adjust the operation of the quantizer so that a predetermined average output is maintained.
- The visual quality of the output can vary from frame to frame as a function of image content.

Image Compression Models

<u>Quantizer:</u> It keeps irrelevant information out of compressed representations.

- This operation is irreversible.
- It must be omitted when error free compression is desired.
- In video applications, bit rate of encoded output is often measured and used to adjust the operation of the quantizer so that a predetermined average output is maintained.
- The visual quality of the output can vary from frame to frame as a function of image content.

Image Compression Models

<u>Quantizer:</u> It keeps irrelevant information out of compressed representations.

- This operation is irreversible.
- It must be omitted when error free compression is desired.
- In video applications, bit rate of encoded output is often measured and used to adjust the operation of the quantizer so that a predetermined average output is maintained.
- The visual quality of the output can vary from frame to frame as a function of image content.

Some Basic Compression Methods

Huffman Coding:

- Most popular technique for removing coding redundancies.
- It yields smallest possible code symbol per source symbol.

Original Source		Source reduction			
Symbol	Probability	1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman Coding:

Original Source			Source reduction			
Symbol	Probability	Code	1	2	3	4
a2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a1	0.1	011	0.1 011	0.2 010	0.3 01	
a4	0.1	0100	0.1 0100	0.1 011		
a3	0.06	01010	0.1 0101			
a5	0.04	01011				

$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.06)(5) + (0.04)(5)$

$= 2.2$ bits / pixel.

Arithmetic coding:

It generates non block codes.

One to One correspondence between source symbols and code words does not exist.

Instead, an entire sequence of source symbols is assigned a single arithmetic code.

Code word defines an integer of real numbers between 0 & 1.

As No. of symbols in msg. ↑
interval to represent it ↓
no. of bits ↑
to represent info ↓

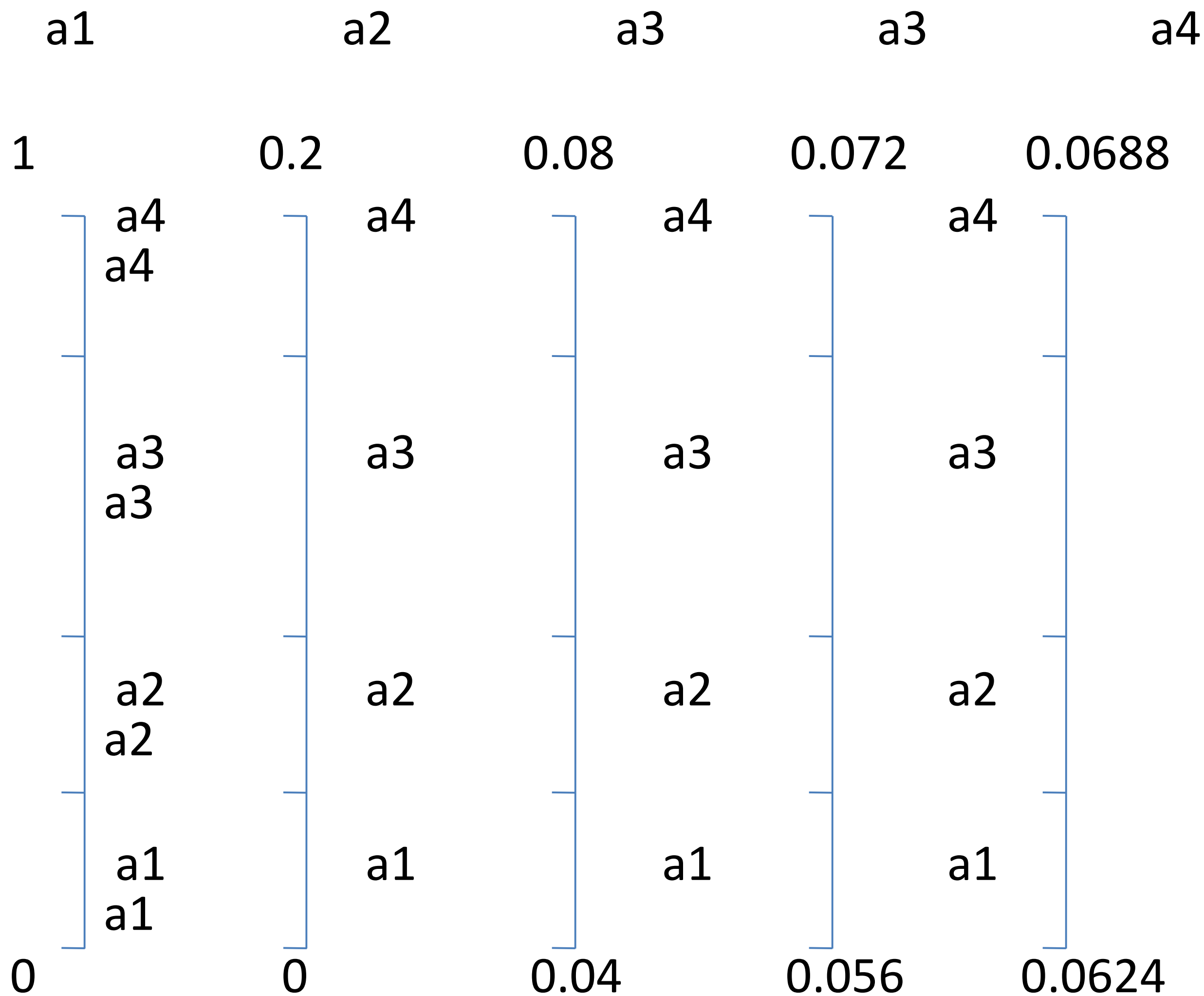
Each symbol of msg size of interval in accordance with its probability of occurrence.

Basic Arithmetic coding process:

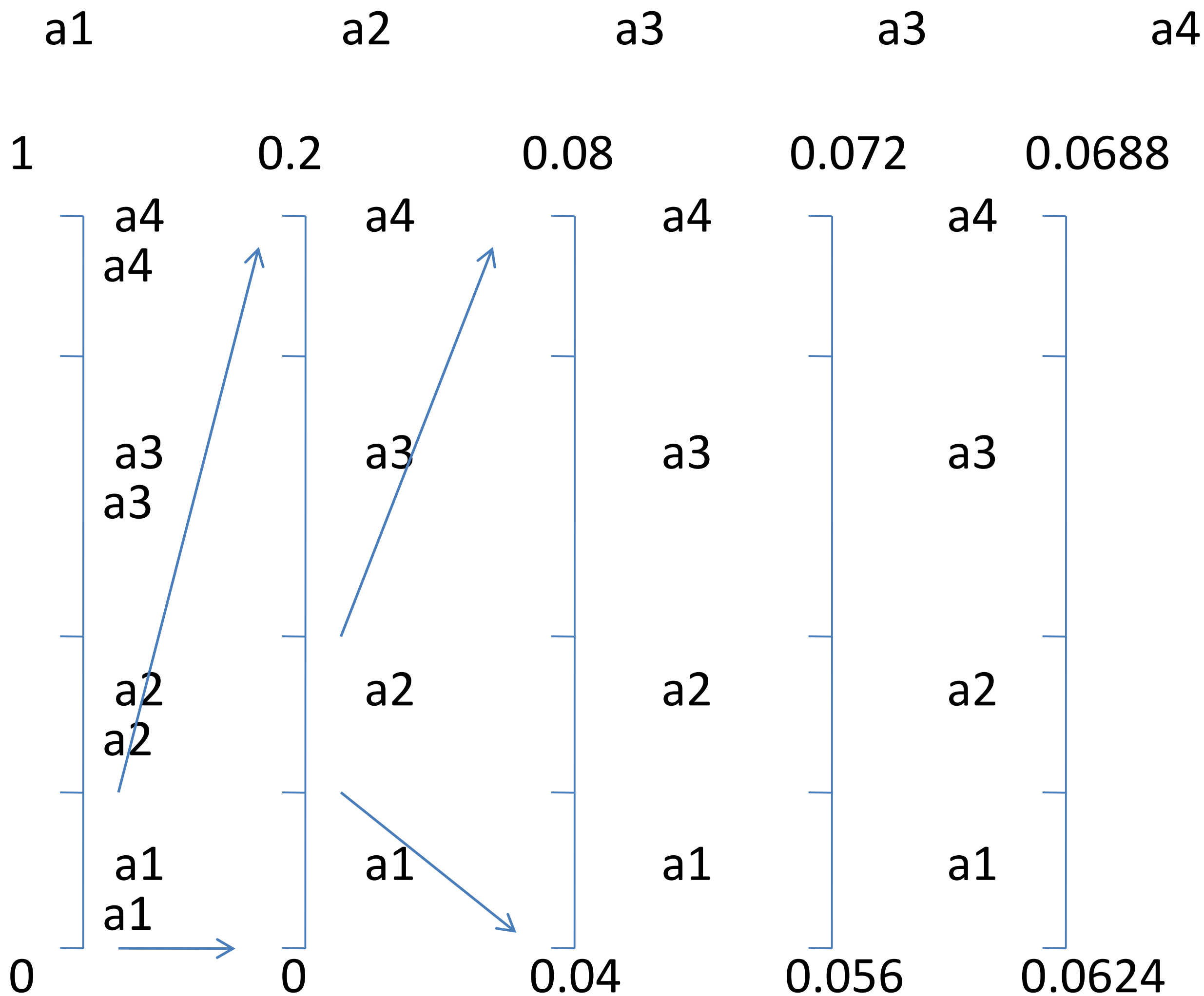
5 symbol message, a1a2a3a3a4 from 4 symbol source is coded.

Source Symbol	Probability	Initial Subinterval
a1	0.2	[0.0, 0.2)
a2	0.2	[0.2, 0.4)
a3	0.4	[0.4, 0.8)
a4	0.2	[0.8, 1.0)

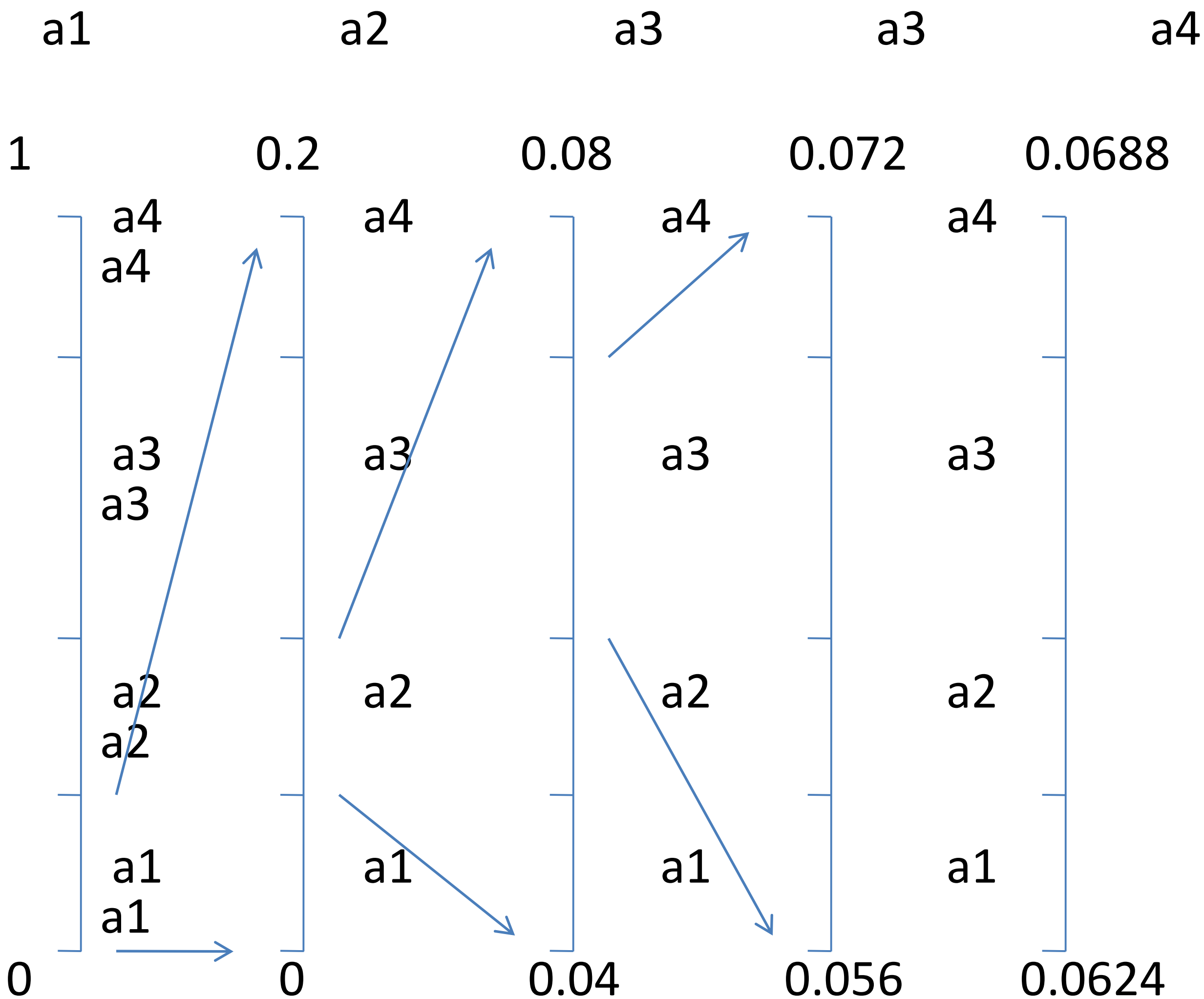
Arithmetic coding:



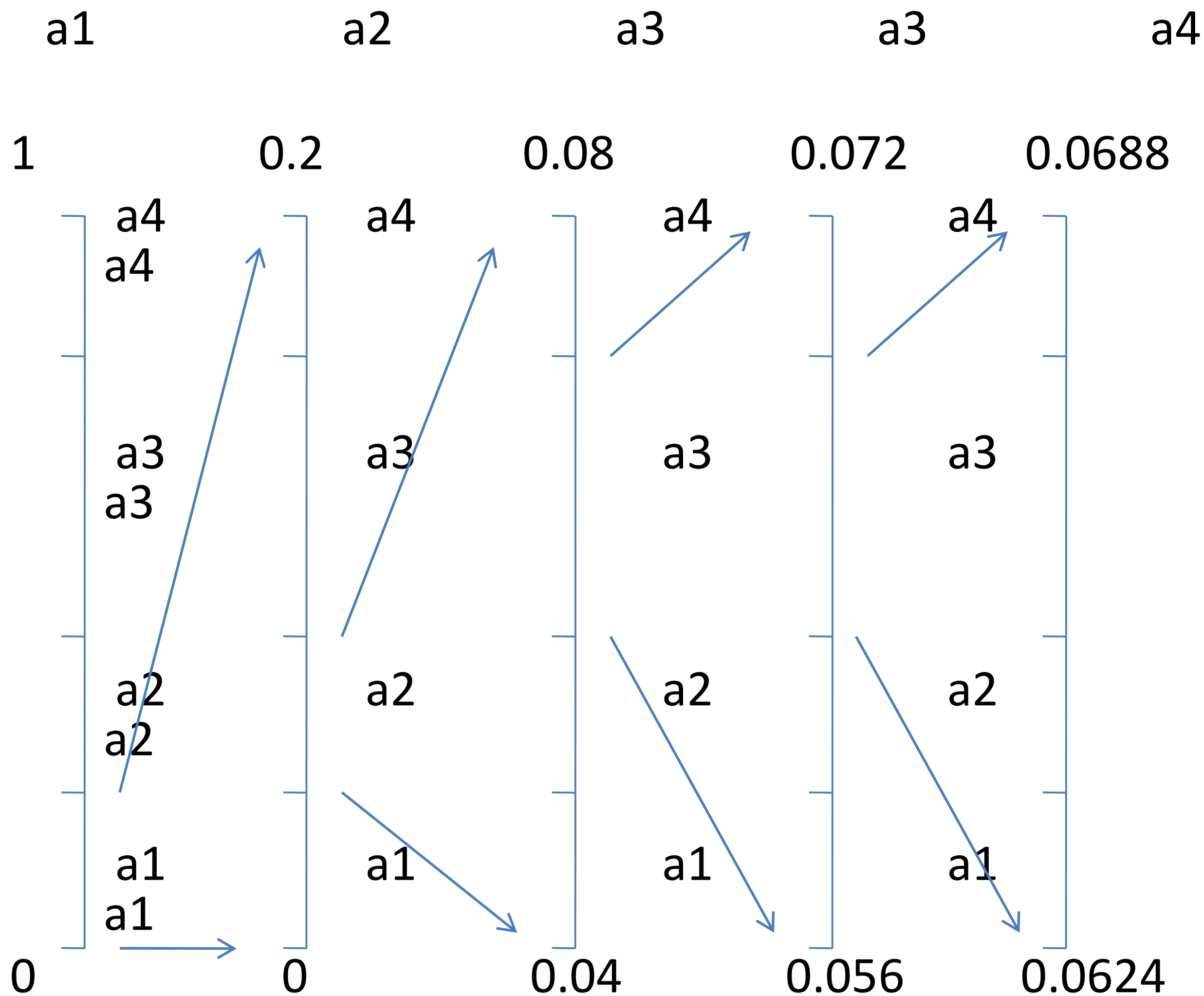
Arithmetic coding:



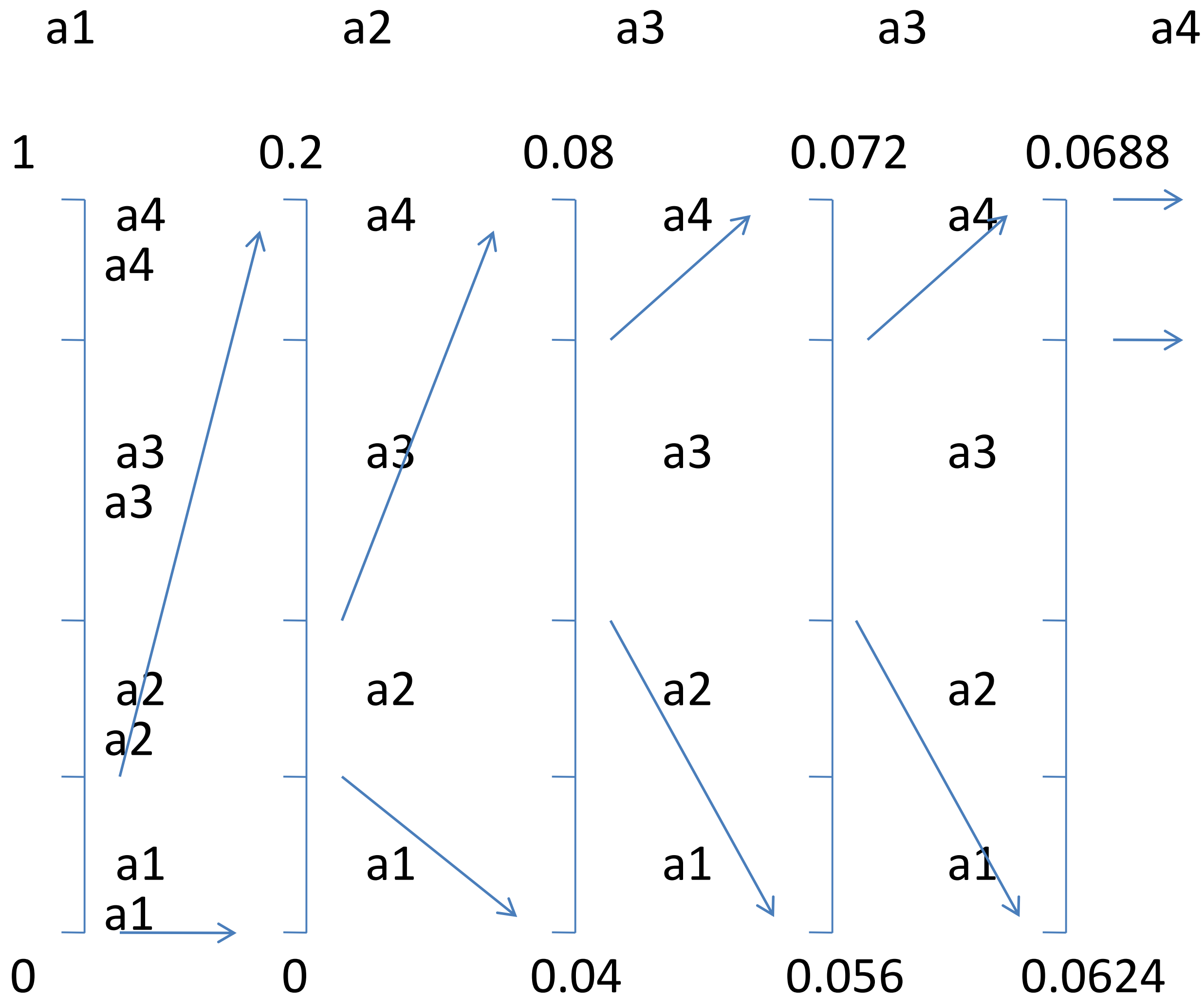
Arithmetic coding:



Arithmetic coding:



Arithmetic coding: Encoding Sequence →



The final message symbol narrows to [0.06752, 0.0688).

Any number between this interval can be used to represent the message.

Eg. 0.068

3 decimal digits are used to represent the 5 symbol message.

LZW Coding:

- LZW – Lempel-Ziv-Welch coding
- Error free compression approach that also addresses spatial redundancies in an image.
- It assigns fixed-length code words to variable length sequences of source symbols.
- It requires no prior knowledge of probability of occurrence of the symbols to be encoded.

LZW Coding:

- It is conceptually very simple.
- Initially, a codebook or dictionary containing the source symbols to be coded is constructed.
- For an 8-bit BW images, the first 256 words of dictionary are assigned to intensities 0, 1, 2, …., 255.

LZW Coding:

Consider the 4 x 4 8-bit image having a vertical edge.

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

A 512-word dictionary with following content is assumed:

Dictionary Location	Entry
0	0
1	1
:	:
255	255
256	-
.	.
511	-

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location	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		

Unique feature of LZW coding:

- Coding dictionary or code book is created while data are being encoded.

Some Basic Compression Methods

Bit-plane coding:

- Another effective method to reduce interpixel redundancies.
- Image's bit planes are processed individually.
- Based on decomposing a multilevel (monochrome / color) image into a series of binary images & compressing each binary using any binary compression method.

Bit plane decomposition:

- Gray levels of an m-bit gray level image can be represented in form of base 2 polynomial.

$$a_{m-1}2^{m-1}+a_{m-2}2^{m-2}+.......+a_1 2^1+a_0 2^0 \quad ....(1)$$

Some Basic Compression Methods

- A simple method of decomposing the image into a collection of binary image is to separate the m coefficients of the polynomial into m-1 bit planes.

Disadvantage:

- Small changes in gray level can have significant impact on complexity of bit planes.

Ex. If two adjacent pixels have intensity of 127 (01111111) and 128 (10000000), every bit plane will contain a corresponding 0 to 1 (or 1 to 0) transition.

Alternate decomposition approach:

- Reduces the effect of small gray level variations.

- Requires the representation of image into m-bit gray code.

- m-bit gray code $g_{m-1}.....g_2 g_1 g_0$ can be computed from

$$g_i = a_i \text{ xor } a_{i+1} \quad 0 \le i \le m-2$$
$$g_{m-1} = a_{m-1}$$

- This code has unique property that successive code words differ by only 1 bit position.

- Small changes in gray level are less likely to affect all m bit planes.

Ex. 127 (11000000) & 128 (01000000).

Run-Length Coding:

- Standard compression approach used in facsimile (FAX).

- Basic concept is to code each contiguous group of 0’s & 1’s encountered in L to R scan of a row by its length & to develop a convention for determining the value of run.

- Most common approach for determining the value of run:

i) Specify value of first run of each row.

ii) To assume each row begins with a white run whose run length may in fact be zero.

- Although Run length coding is in itself an effective method of compressing an image, additional compression can be realized by variable-length coding.

- Black & white run lengths may be coded separately using variable-length codes.

Ex. If a_j represent a black run of length j, we can estimate its probability.

- The approximate run-length entropy of the image is:

$$H_{RL} = (H_0 + H_1) / (L_0 + L_1)$$

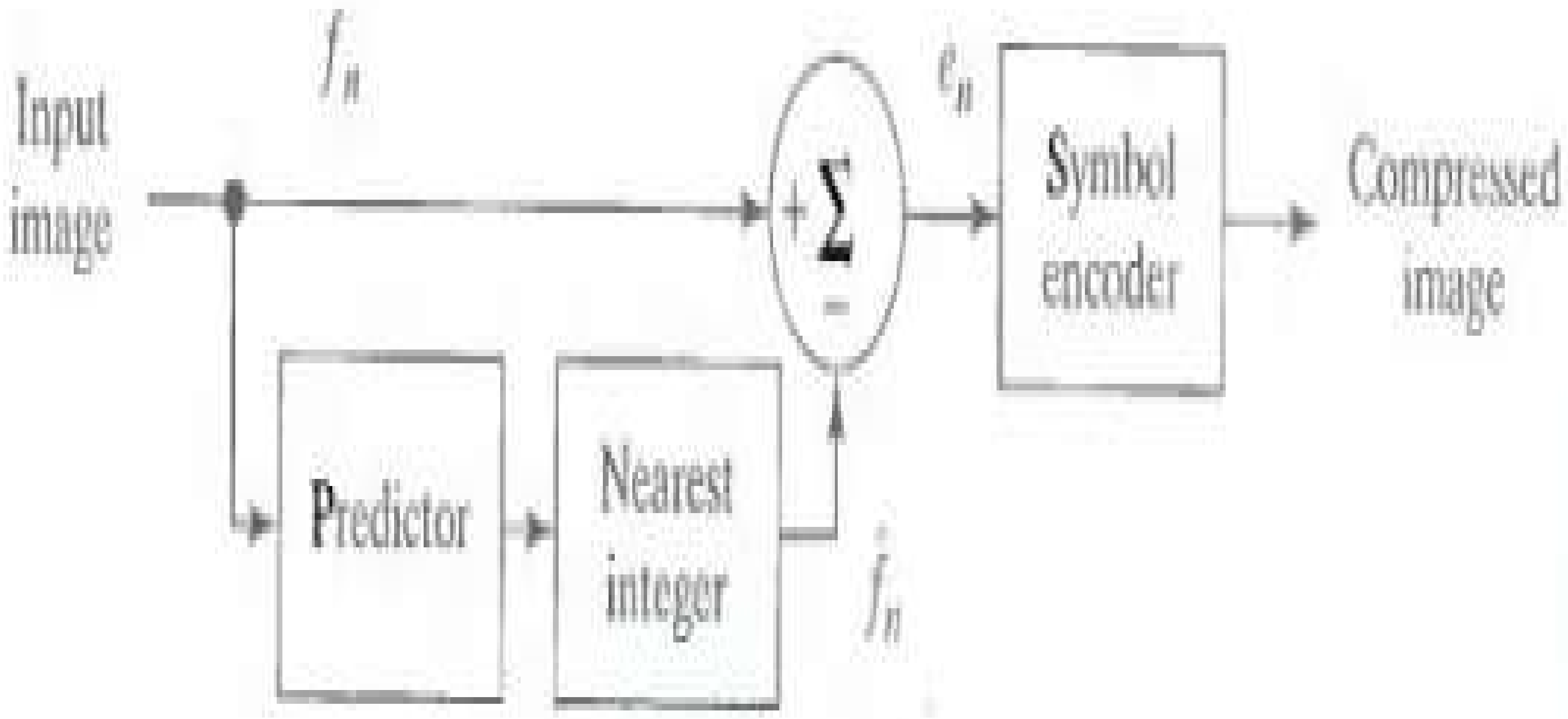
- where, L_0 & L_1 denote average values of black & white run lengths resp.

- Above equation also provides an estimate of the average no. of bits per pixel required to code the run lengths in a binary image.

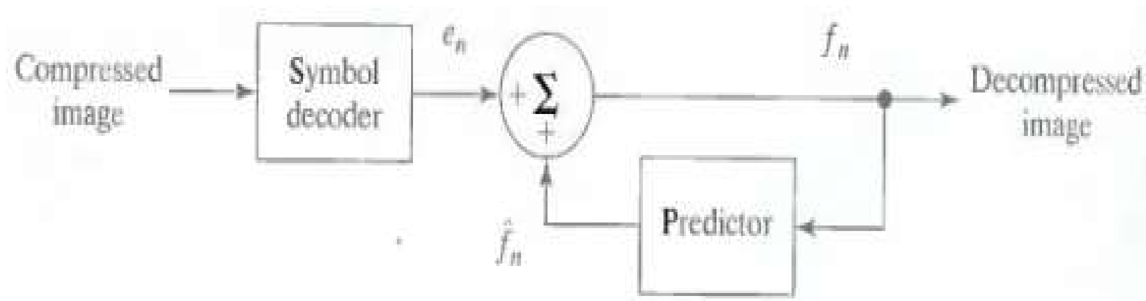
Lossless predictive coding:

- Based on eliminating the interpixel redundancies of closely spaced pixels by extracting & coding only the new information in each pixel.

- New information: difference between the actual & predicted value of that pixel.


Input image
f_n
e_n
Symbol encoder
Compressed image
Predictor
Nearest integer
$\hat{f}_n$

- Figure shows basic component of a lossless predictive coding system.



- It consists of an encoder & a decoder each containing an identical predictor.

- As each successive pixel of input image f(n) is introduced to the encoder, predictor generates its anticipated value.

Output of the predictor is then rounded to the nearest integer f(n)bar & used to form the difference or prediction error.

- It is coded using a variable length to generate the next element of the compressed data stream.

- The decoder reconstruct the fn from the received variable-length code words & perform the inverse operation

- f(n) = e(n) + f(n)bar

- f(n)bar is generated by prediction formed by a linear combination of m previous pixels.

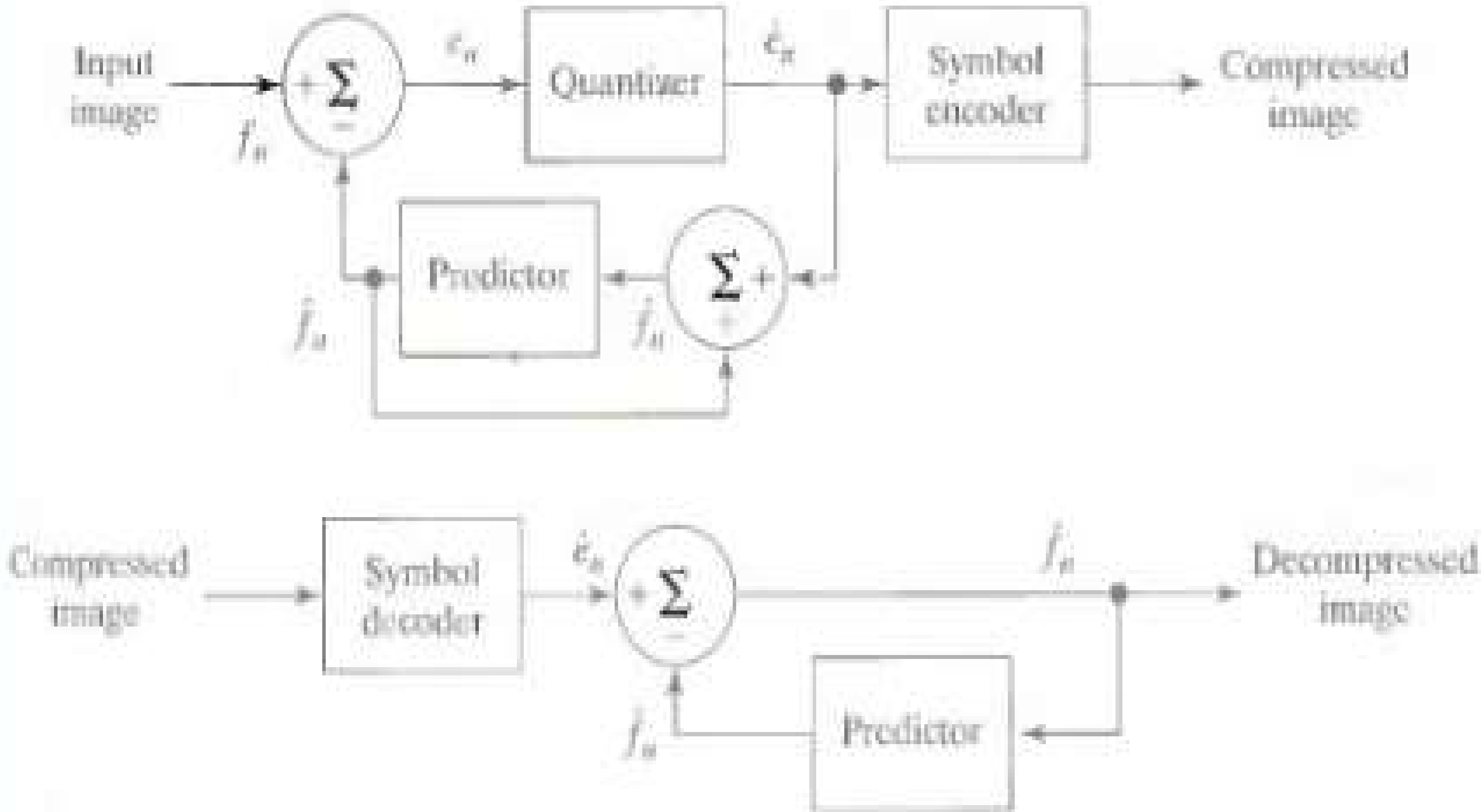
$$\bar{f}(n) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(n-i)\right]$$

where, m – order of linear predictor

round – function used to denote rounding

α_i – for i = 1, 2, 3, ….. m are prediction coefficients.

Lossy Predictive coding:



Lossy Predictive coding:

- In this method we add a quantizer to the lossless predictive coding model.

- It replaces the nearest integer function & is placed between symbol encoder & point where prediction error forms.

- It maps the prediction error into a limited range of outputs denoted by ë(n), which establish the amount of compression & distortion.

- In order to accommodate the insertion of the quantization step, the error free encoder must be altered so that the predictions by the encoder & decoder are equivalent.

Lossy Predictive coding:

- This is accomplished by placing the predictor within a feedback loop, where its input ƒ(n)dot is generated as a function of past predictions & quantized errors.

ƒ(n)dot = e(n)dot + f(n)bar

- This closed loop configuration prevents error buildup at the decoder's output.

Chapter 6
Color Image Processing

Isaac Newton, 1666

FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Recall that the 6-7 million cones (sensors) in the human eye are responsible for color vision, see Chapter 2.

Experimental evidence shows that theses can be divided into three principal sensing categories corresponding to roughly red, green and blue: (65% of cones are sensitive to RED, 33% to GREEN and 2% to blue)

We, therefore, have three brightness response functions:

$$f_R(x,y) = \int_0^\infty C(x,y,\lambda)\, V_R(\lambda)\, d\lambda$$

$$f_G(x,y) = \int_0^\infty C(x,y,\lambda)\, V_G(\lambda)\, d\lambda$$

$$f_B(x,y) = \int_0^\infty C(x,y,\lambda)\, V_B(\lambda)\, d\lambda$$

The three relative luminous efficiency functions are plotted in the next slide.

Color Image Processing

1965 Experimental curves:

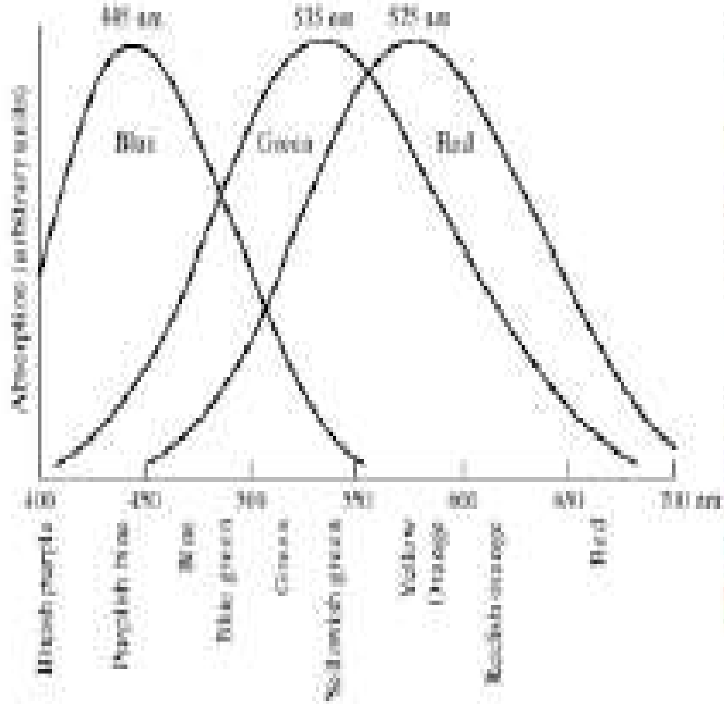


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

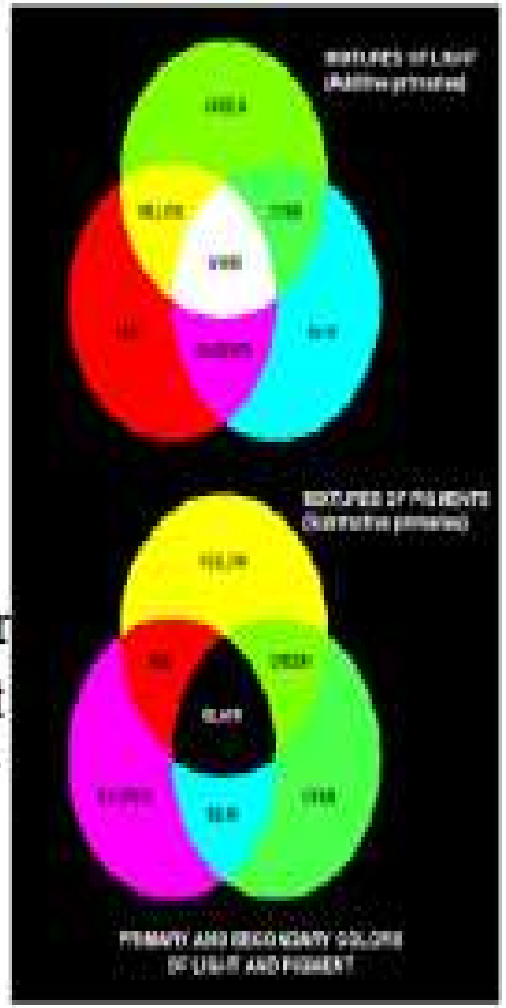
Due to these absorption characteristics, colors are seen as variable combinations of so called "primary" colors red, green and blue.

In 1931, CIE designated the following:
Blue = 435.8nm;
Green = 546.1nm; and
Red = 700nm

- Remember that there is no single color called red, green or blue in the color spectrum!
- Also, these fixed RGB components cannot generate ALL spectrum colors!

Primary colors can be added in pairs to procude secondary colors of light: e.g. magenta, cyan and yellow.
Mixing the three primaries produces white color.

A primary color of pigments or colorants is defined as one that Subtracts or absorbs a primary color of light and reflects the other two.



primary colors of pigments are magenta, cyan and yellow and their secondary colors are red, green and blue

FIGURE 6.4 Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lamp Business Division.)

Tristimulus Values (X,Y,Z)

Are the amounts of red, green and blue needed to form any particular color.

A color is specified by its trichromatic coefficients defined as:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z}$$

Note that $x+y+z = 1$! (i,e. only two of the trichromatic coefficients are independent.)

Experimental curves and tables are used to find the tristimulus values needed to generate a given color.

Color models or color spaces refer to a color coordinate system in which each point represents one color.

Different models are defined (standardized) for different purposes, e.g.

Hardware oriented models:

- RGB for color monitors (CRT and LCD) and video cameras,
- CMYK (cyan, magenta, yellow and black) for color printers

Color manipulation models:

- HSI (hue, saturation and brightness) is closest to the human visual system
- Lab is most uniform color space
- YCbCr (or YUV) is often used in video where chroma is down-sampled (recall that the human visual system is much more sensitive to luminance than to color)
- XYZ is known as the raw format
- others

Two important aspects to retain about color models:

1. conversion between color models can be either linear or nonlinear,
2. some models can be more useful as they can decouple color and gray-scale components of a color image, e.g. HSI, YUV.

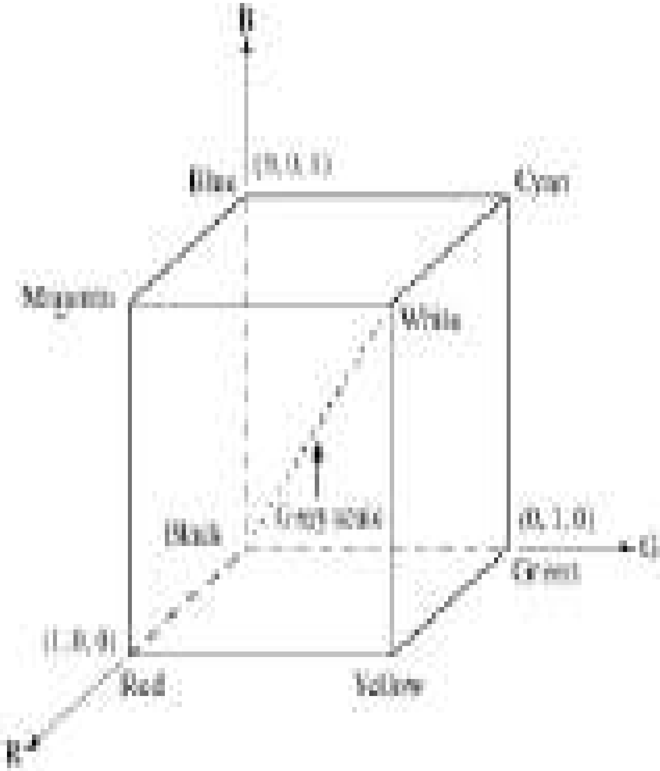
Three Perceptual Measures

1. **Brightness**: varies along the vertical axis and measures the extent to which an area appears to exhibit light. It is proportional to the electromagnetic energy radiated by the source.

2. ***Hue***: denoted by H and varies along the circumference. It measure the extent to which an area matches colors red, orange, yellow, blue or purple (or a mixture of any two). In other words, hue is a parameter which distinguishes the color of the source, i.e., is the color red, yellow, blue, etc.

3. **Saturation**: the quantity which distinguishes a pure spectral light from a pastel shade of the same hue. It is simply a measure of white light added to the pure spectral color. In other words, saturation is the

FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).



The eight vertices of the cube are occupied by

red, green and blue;

magenta, cyan and yellow;

and finally black and white.

(RGB values have been normalised in the range [0,1])

Pixel depth refers to the number of bits used to represent each pixel in the RGB space

If each pixel component (red, green and blue) is represented by 8 bits, the pixel is said to have a depth of 24 bits.

A **full-color** image refers to a 24-bit RGB color image. The number of possible colors in a full-color image is:

$$(2^8)^3 = 16{,}777{,}216 \text{ colors (or 16 million colors)}$$

CMY and CMYK Color Models

Most devices that deposit color pigments on paper, e.g. printers and copiers, use CMY inputs or perform RGB to CMY conversion internally:

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

Recall that all color values have been normalised in the range [0,1].

Remarks:

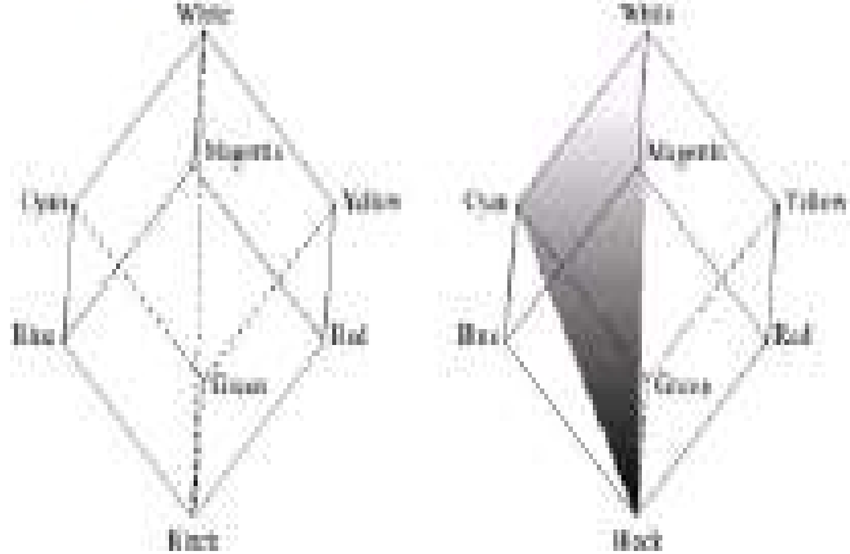
1. Note that, e.g. a surface coated with cyan does not contain red, that is C = 1 – R.

2. Since equal amounts of the pigment primaries should produce black, in printing this appears as muddy-looking black; therefore, a fourth color, black is added, leading to CMYK color model (four-color printing).

HSI Color Model

- Although RGB and CMY color models are very well suited for hardware and RGB reflects well the sensitivity of the human eye to these primary colors, both are not suited for describing color in a way that is easily interpreted by human.
- When human see a color object, they tend to describe it by its hue, saturation and brightness, i.e. HSI model is used
- In addition, HSI decouples brightness from the chroma components.

Perceptual relationship between RGB and HSI color models



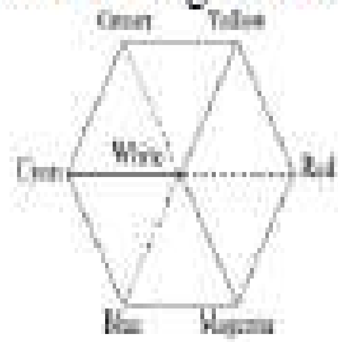
a b

FIGURE 6.13 Conceptual relationships between the RGB and HSI color models.

- Note that the intensity increases from black to white
- All points along the intensity axis are gray and thus have 0 saturation value.
- Saturation increases as a function of the distance from the intensity axis
- The shaded region has a single color, cyan, with different shades, rotating it wrt to intensity axis results in a new hue value (new color)

In fact, HSI is represented by a vertical intensity axis and the locus of color points lying on planes ⊥ to this axis. The boundary of these planes defined by the intersection with the faces of the cube is either hexagonal or triangular, see below.

Note also that primaries are separated by 120



For visualization purposes, can also display the boundary as a circle!

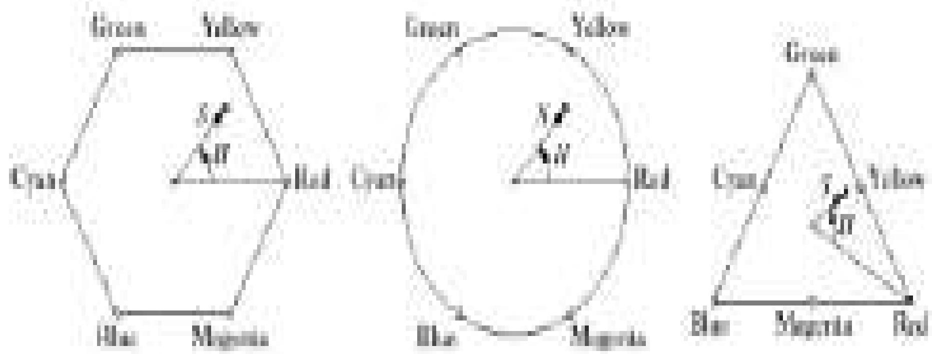


FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

HSI-RGB Color Model Conversions

From RGB to HSI:

the hue is: $$H = \begin{cases} \theta & \text{if} \quad B \le G \\ 360 - \theta & \text{if} \quad B > G \end{cases}$$

with $$\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}} \right\}$$

the saturation: $$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

and the intensity is: $$I = \frac{1}{3}(R+G+B)$$

It's assumed that RGB values are normalised and Θ is measured wrt red axis see Fig. 6.13.

Conversion from HSI to RGB depends on which sector H is located, see details in Eqs. 6.2-5 – 6.2-15.

IMAGE SEGMENTATION

Introduction

- Segmentation refers to another step in image processing methods where input are images and outputs are attributes extracted from images.
- It subdivides an image into its constituent regions or objects.
- Segmentation accuracy determines the eventual success or failure of computerized analysis procedures.
- Ex. Autonomous target acquisition
- Segmentation algorithms are based on 1 of 2 basic properties of intensity values: discontinuity & similarity.

Introduction

- Segmentation algorithms are based on 1 of 2 basic properties of intensity values: discontinuity & similarity.

- Discontinuity: Approach is to partition image based on abrupt changes in intensities (edges).

- Similarity: Approach is to partition the image based on similar regions according to predefined criteria.

- Such as Thresholding, region growing, region splitting & merging.

Fundamentals

- Let R represent the entire region occupied by an image.
- Image segmentation partitions R into n subregions R_1, R_2, ….., R_n, such that

a) U R_i = R

b) R_i is a connected set, i = 1, 2, …., n

c) R_i n R_j = ø for all i & j, i ≠ j

d) $Q(R_i)$ = TRUE for i = 1, 2, ….., n

e) $Q(R_i \cup R_j)$ = FALSE for any adjacent regions R_i & R_j.

Fundamentals

$Q(R_k)$ is a logical predicate over the points in set Rk

Explanation:

a) Every pixel must be in a region.

b) Points in a region be 4- or 8- connected.

c) Regions must be disjoint

d) $Q(R_i)$ = TRUE if all pixels have same intensity level.

e) Two adjacent regions R_i & R_j must be different in the sense of predicate Q.

Point, Line & Edge Detection

3 type of image features:

- Points
- Line
- Edges

- Edge Pixels: pixels at which intensity of an image changes abruptly.

- Edges (Edge Segments): Are set of connected edge pixels.

- Local Averaging smoothens an image.
- Averaging is analogous to Integration.
- Local changes in intensities can be detected using derivatives.
- First & Second order derivative more suitable.

Point, Line & Edge Detection

- Derivatives of a digital function are defined in terms of differences.

An approximation used for first derivative

(i) Must be zero in areas of constant intensity

(ii) Must be non-zero at the onset of an intensity step or ramp.

(iii) Must be nonzero at points along an intensity ramp.

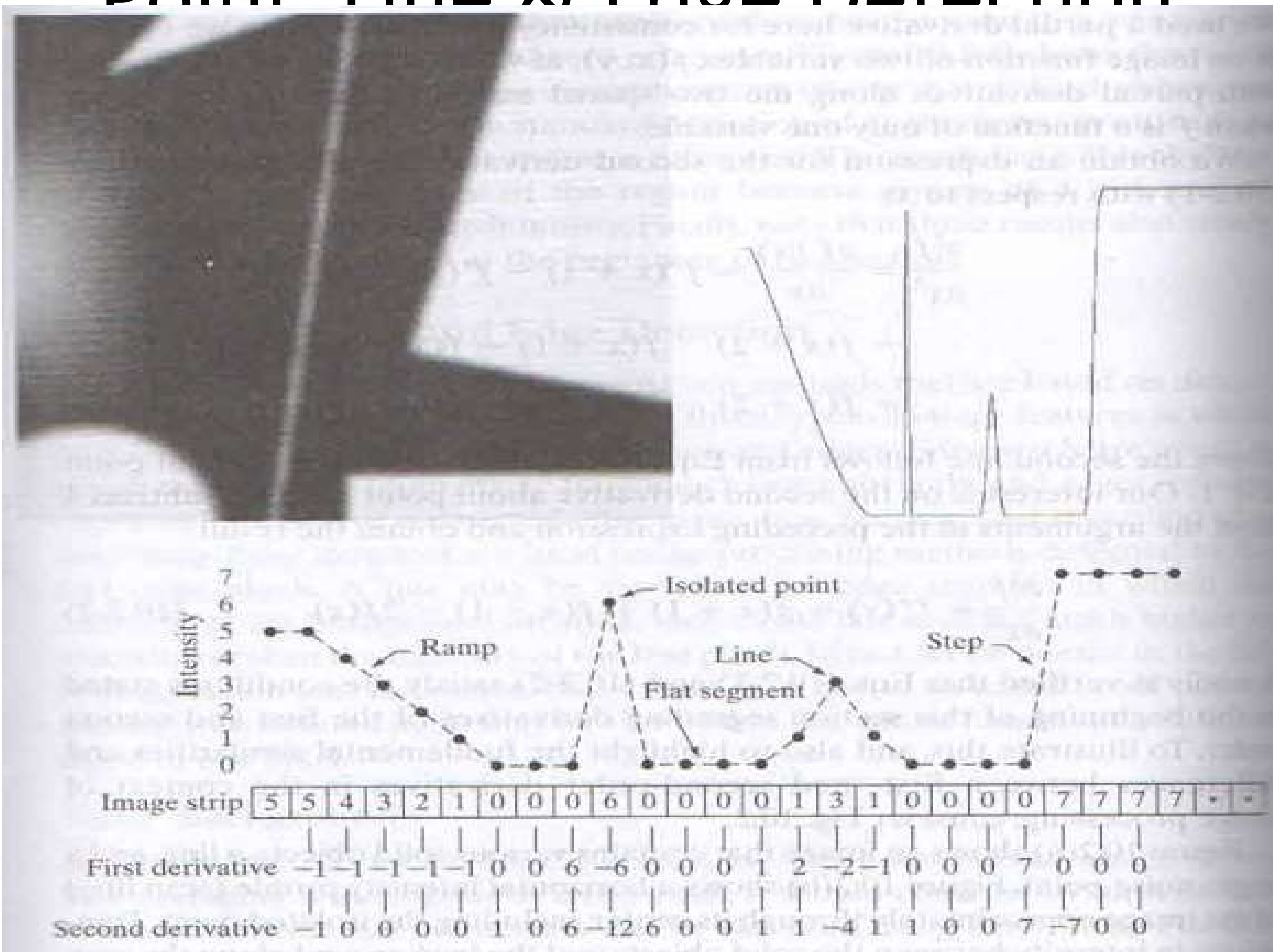
An approximation used for Second derivative

(i) Must be zero in areas of constant intensity

(ii) Must be non-zero at the onset & end of an intensity step or ramp.

(iii) Must be zero along an intensity ramps.

Point, Line & Edge Detection



Point, Line & Edge Detection

- Consider the properties of first & second derivatives as we traverse from left to right.

- Initially we note that the first-order derivative is non-zero at the onset & along the entire intensity ramp.

- While the second order derivative is non-zero only at the onset and end of the ramp.

Point, Line & Edge Detection

We conclude:

1) First order derivative produce “thick” edges
2) Second order derivative have stronger response to “finer” details eg. Thin lines, isolated points & noise.
3) Second order derivative produce a double-edge response at ramp & step transitions in intensity.
4) The sign of the second derivative can be used to determine whether a transition into an edge is from Light to Dark or vice versa.

Spatial filters can fulfill the requirements of first & second order derivatives.

Point, Line & Edge Detection

- For a 3 x 3 filter mask, the procedure is to compute the sum of products of the mask coefficients with the intensity values in the region encompassed be the mask.
- Response of the mask at the center point of the region is:

$$R = w_1z_1 + w_2z_2 + ... + w_9z_9$$

$$= \sum_{k=1}^{9} w_kz_k$$

Where, z_k is the intensity of k^{th} pixel

w1	w2	w3
w4	w5	w6
w7	w8	w9

Point, Line & Edge Detection

Point Detection:

It should be based on the second derivative.

Using the Laplacian

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

Where the partials are given by

$$\partial^2 f(x, y)/\partial x^2 = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

&

$$\partial^2 f(x, y)/\partial y^2 = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

Laplacian is then given by:

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

Point, Line & Edge Detection

- Above expression can be implemented using mask shown earlier

- Using Laplacian mask, we say a point at location (x, y) is detected if
- Absolute value of the response of the mask at that point exceeds a threshold.
- Such points are labeled 1 in output image & others 0, thus producing binary image.

$$g(x, y) = \begin{cases} 1 & \text{if } |R(x, y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$

✓ !dea : Intensity of an isolated point will differ from surrounding thus easily detected.

Point, Line & Edge Detection

Line Detection:

- We can expect second derivatives to result in a stronger response & to produce thinner lines than first derivatives.
- Thus we use same Laplacian mask for line detection as well.
- But the mask is isotropic, so its response is independent of direction (wrt 4 dimensions of 3x3 Laplacian mask: vertical , horizontal & 2 diagonals.)
- We may need to determine line in a specific direction.

Point, Line & Edge Detection

Line Detection masks:

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

Horizontal

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

+ 45°

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

Vertical

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

- 45°

Point, Line & Edge Detection

- Let R1, R2, R3 & R4 denote responses of the masks from Horizontal, +45°, Vertical & -45°.

- If at a given point in the image, $|R_k| > |R_j|$ for all $j \neq k$, that point is said to be more likely associated with a line in the direction of mask k.

Point, Line & Edge Detection

Edge Models:

- It is the approach used most frequently for segmenting images based on abrupt (local) changes in intensity.

Edge Models are classified according to their intensity profiles

Step Edge:

- It involves a transition between two intensity levels occurring ideally over the distance of 1 pixel.

Eg. Computer generated images

Ramp Edge:

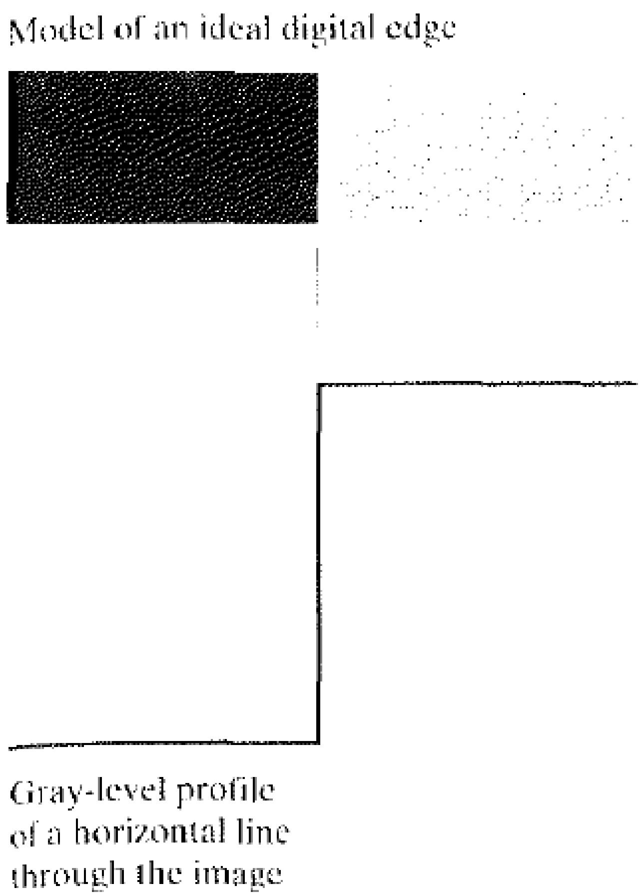
- Digital images with edges blurred with noise, with the degree of blurring determined by limitations in focusing mechanism.
- Slope of ramp is inversely proportional to the degree of blurring in the edge.

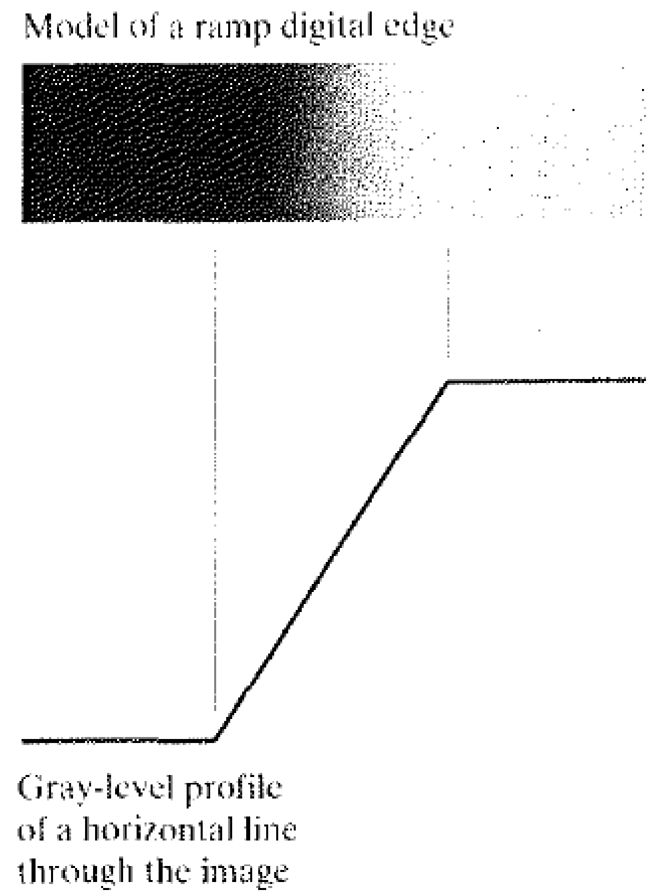
Point, Line & Edge Detection

Roof Edge:

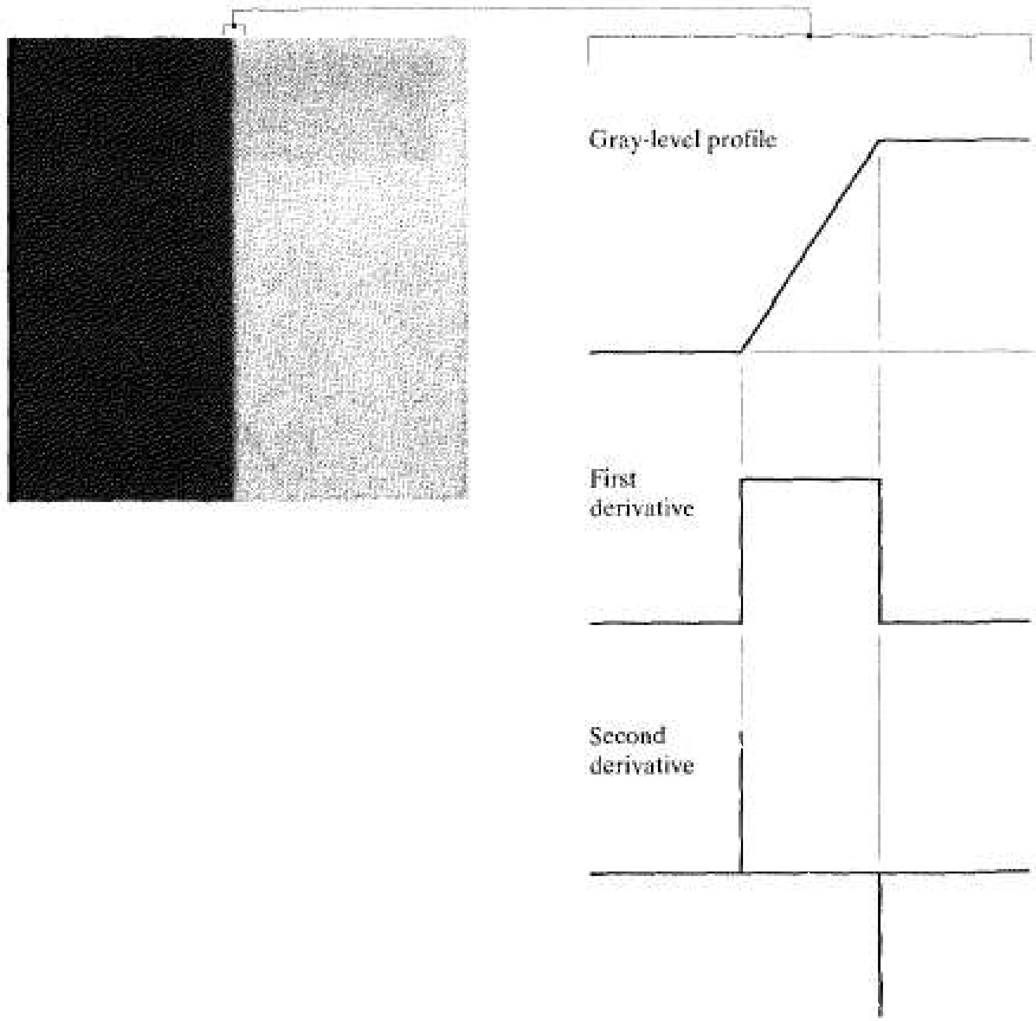
- Roof Edges are models of lines through a region, with the base (width) of a roof edge being determined by the thickness & sharpness of the line.
- Eg. In range imaging, when thin objects (pipe) are closer to the sensor, pipes appear brighter.
- Digitization of lines drawn, in satellite imaging roads look like as shown in image.

Point, Line & Edge Detection





Point, Line & Edge Detection



Point, Line & Edge Detection

Conclusion :

- Magnitude of the first derivative can be used to detect the presence of an edge at a point in an image.
- Sign of second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge.

Additional Features:

- It provides two values for every edge in an image (undesirable feature).
- Zero crossings can be used for locating the centers of thick edges.

Point, Line & Edge Detection

Conclusion of Section:

1) Image smoothing for noise reduction

2) Detection of edge points: Local operation that extracts all points from an image that are potential candidates to become edge points.

3) Edge localization: To select from the candidate edge points only that are true members of the set of points comprising an edge.

Point, Line & Edge Detection

Edge Linking & Boundary Detection:

- Ideally edge detection should yield sets of pixels only on edges.
- Actually, these pixels seldom characterize edges completely because of noise, breaks in edges due to non uniform illumination, & other effects that introduce spurious discontinuities in intensity values.
- So, Edge detection is followed by edge linking algorithm.

Point, Line & Edge Detection

Local Processing:

- One of the simplest approach for edge linking is to analyze the characteristics of pixels in a small neighborhood about every point (x, y) that has been declared an edge point.

- Two principal properties used for establishing similarity of edge pixels in this kind of analysis are:

1) The strength (magnitude)
2) the direction of the gradient vector.

- Let S_{xy} denote the set of coordinates of a neighborhood centered at point (x, y) in an image.
- An edge pixel with coordinates (s, t) in S_{xy} is similar in magnitude to the pixel at (x, y) if

$$|M(s, t) - M(x, y)| \le E$$

Where E is a positive threshold

Point, Line & Edge Detection

- The direction angle of the gradient vector is given below.

- An edge pixel with coordinates (s, t) in S_{xy} has an angle similar to the pixels at (x, y) if

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

Where A is a positive angle threshold

- A pixel with coordinates (s, t) in S_{xy} is linked to the pixel at (x, y) if both magnitude & direction criteria are satisfied.

Point, Line & Edge Detection

Global Processing using the Hough transform:

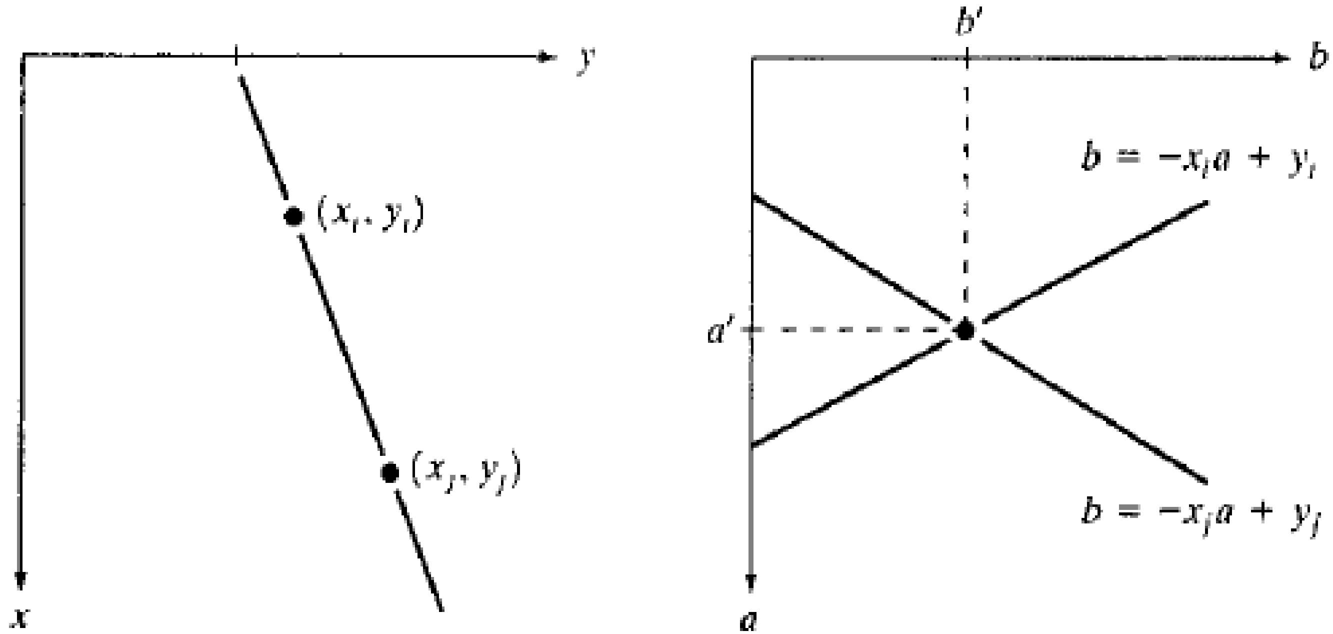
- Here, we develop an approach based on whether sets of pixels lie on curves of a specified shape. Once detected, these curves form the edges or region boundaries of interest.

- Given n points in an image. Suppose that we want to find subset of those points that lie on straight lines.

- One solution is to find all lines determined by every pair of points & then find all subsets of points that are close to particular lines.

- This approach involves finding n(n-1)/2 ≈ n2 lines & then performing n(n(n-1))/2 ≈ n3 comparisons of every point to all lines.

Point, Line & Edge Detection



Point, Line & Edge Detection