Multimedia Systems
Lecture – 10

By

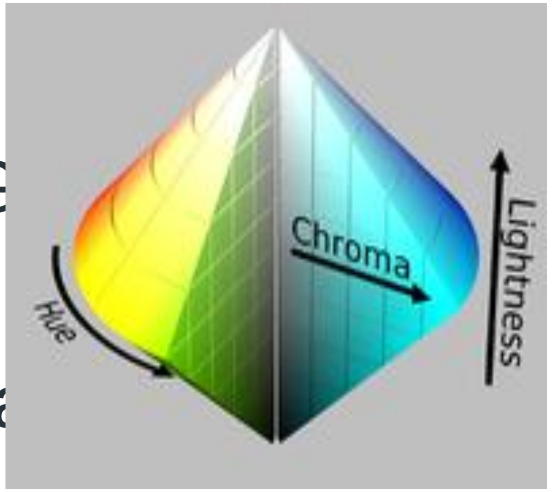
Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

HSV model

- It generalizes how humans perceive color.
- How humans perceive colors is not like how RG
  CMYK make colors.
- The H stands for Hue, S stands for Saturation, a
  the V stand for value.
- Imagine a cone with a spectrum of red to blue from left to right, and from the centre to the edge, the color intensity increases.
- From bottom to up, the brightness increases. Hence resulting white at the center up layer.



- **Hue**: The hue represents the color. The hue value ranges from o to 360 degrees.

- **Saturation**: The saturation value tells us how much quantity of respective color must be added.
- A 100% saturation means that complete pure color is added, while a 0% saturation means no color is added, resulting in grayscale.

- **Value**: The value represents the brightness concerning the saturation of the color. The value 0 represents total black darkness, while the value 255 will mean a full brightness and depend on the saturation.

RGB to HSV

$$M = \max\{R, G, B\}$$

$$m = \min\{R, G, B\}$$

$$V = M$$

$$S = \begin{cases} 0 & \text{if } V = 0 \\ \left(V - m\right)/V & \text{if } V > 0 \end{cases}$$

$$H = \begin{cases} 0 & \text{if } S = 0 \\ 60(G - B)/(M - m) & \text{if } (M = R \text{ and } G \geq B) \\ 60(G - B)/(M - m) + 360 & \text{if } (M = R \text{ and } G < B) \\ 60(B - R)/(M - m) + 120 & \text{if } M = G \\ 60(R - G)/(M - m) + 240 & \text{if } M = B \end{cases}$$

HSV to RGB

- Given the values of H, S, and V, you can first compute m and M with the equations

$$M = V$$

$$m = M(1-S).$$

- Now compute another number, z, defined by the equation

$$z = (M-m)[1 - |(H/60)\mathrm{mod\_2} - 1|],$$

where mod_2 means division modulo 2.

- Now you can compute R, G, and B according to the angle measure of H.
- There are six cases.

- When $0 \le H < 60$,

  $R = M$

  $G = z + m$

  $B = m$.

- If $60 \le H < 120$,

  $R = z + m$

  $G = M$

  $B = m$.

- If $120 \le H < 180$,

  $R = m$

  $G = M$

  $B = z + m$.

- When $180 \le H < 240$,

  $R = m$

  $G = z + m$

  $B = M$.

- When $240 \le H < 300$,

  $R = z + m$

  $G = m$

  $B = M$.

- if $300 \le H < 360$,

  $R = M$

  $G = m$

  $B = z + m$.

Popular Image File Formats

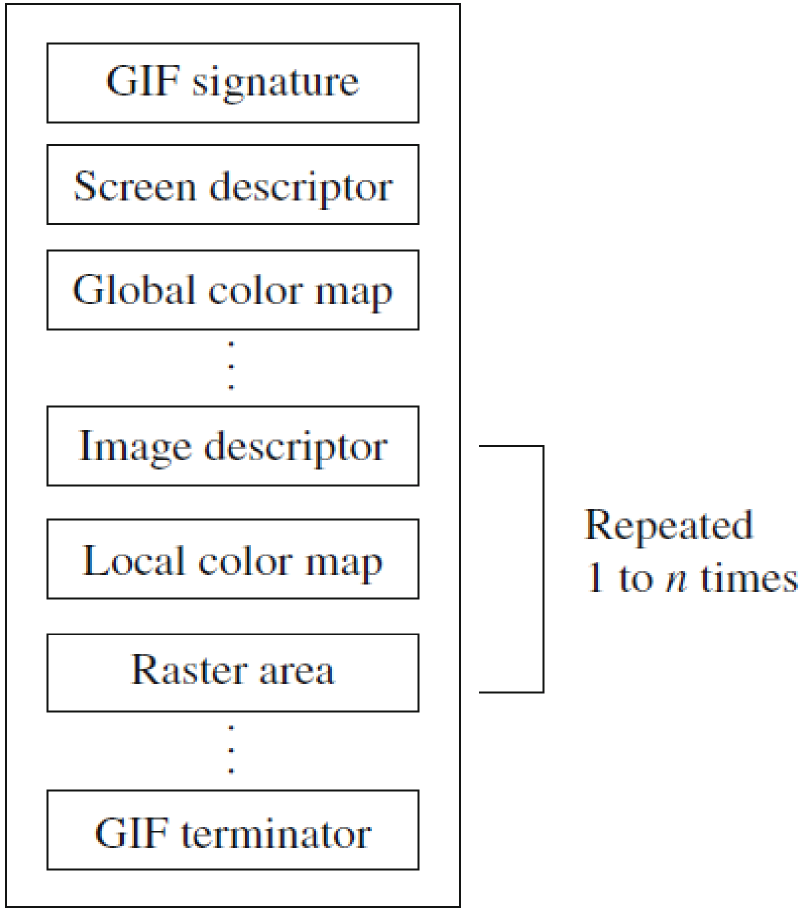
- 8-bit GIF: one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.
- JPEG: currently the most important common file format.
- PNG: most popular lossless image format.
- TIFF: flexible file format due to the addition of tags.
- EXIF: allows the addition of image metadata.
- PS and PDF: vector based language, popular in publishing and academia

GIF (Graphic Interchange Format)

- GIF standard: We examine GIF standard because it is so simple! yet contains many common elements.
- Limited to 8-bit (256) color images only, which, while producing acceptable color images, is best suited for images with few distinctive colors (e.g., graphics or drawing).
- GIF standard supports interlacing — successive display of pixels in widely-spaced rows by a 4-pass display process.
- GIF actually comes in two flavors:
  1. GIF87a: The original specification.
  2. GIF89a: The later version. Supports simple animation via a Graphics Control Extension block in the data, provides simple control over delay time, a transparency index, etc.

GIF87

- Since many such formats bear a resemblance to it but have grown a good deal more complex than this “simple” standard, it is worth examining the file format for GIF87 in more detail.
- For the standard specification, the general file format is as shown in the figure.

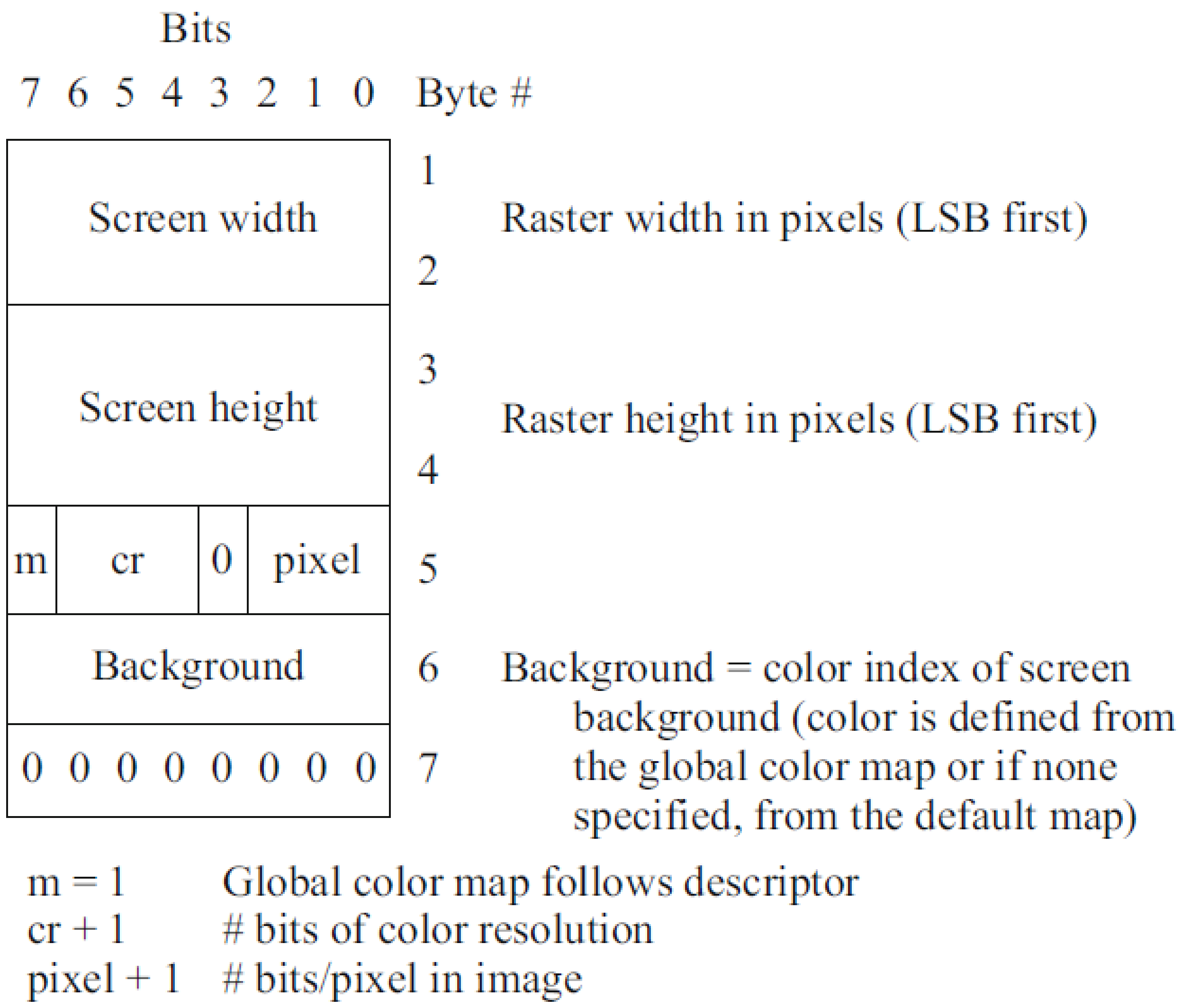


- The *Signature* is six bytes: GIF87a; the *Screen Descriptor* is a seven-byte set of flags.

- A GIF87 file can contain more than one image definition, usually to fit on several different parts of the screen.

- Therefore each image can contain its own color lookup table, a *Local Color Map*, for mapping 8 bits into 24-bit RGB values.

- The Screen Descriptor comprises a set of attributes that belong to every image in the file.



GIF Color Map

Bits 7 6 5 4 3 2 1 0	Byte #	
Red intensity	1	Red value for color index 0
Green intensity	2	Green value for color index 0
Blue intensity	3	Blue value for color index 0
Red intensity	4	Red value for color index 1
Green intensity	5	Green value for color index 1
Blue intensity	6	Blue value for color index 1
⋮		(continues for remaining colors)

GIF image descriptor

Bits 7 6 5 4 3 2 1 0	Byte #	
0 0 1 0 1 1 0 0	1	Image separator character (comma)
Image left	2 3	Start of image in pixels from the left side of the screen (LSB first)
Image top	4 5	Start of image in pixels from the top of the screen (LSB first)
Image width	6 7	Width of the image in pixels (LSB first)
Image height	8 9	Height of the image in pixels (LSB first)
m | i | 0 | 0 | 0 | pixel	10	m = 0 Use global color map, ignore 'pixel' m = 1 Local color map follows, use 'pixel' i = 0 Image formatted in Sequential order i = 1 Image formatted in Interlaced order pixel + 1 # bits per pixel for this image

Raster Area:

- The format of the actual image is defined as the series of pixel color index values that make up the image.
- The pixels are stored left to right sequentially for an image row.
- By default each image row is written sequentially, top to bottom.
- In the case that the Interlace or ‘i' bit is set in byte 10 of the Image Descriptor then the row order of the image display follows a four-pass process in which the image is filled in by widely spaced rows.
- The first pass writes every 8th row, starting with the top row of the image window. The second pass writes every 8th row starting at the fifth row from the top. The third pass writes every 4th row starting at the third row from the top. The fourth pass completes the image, writing every other row, starting at the second row from the top.

GIF four-pass interlace display row order

Image row	Pass 1	Pass 2	Pass 3	Pass 4	Result
0	*1a*				*1a*
1				*4a*	*4a*
2			*3a*		*3a*
3				*4b*	*4b*
4		*2a*			*2a*
5				*4c*	*4c*
6			*3b*		*3b*
7				*4d*	*4d*
8	*1b*				*1b*
9				*4e*	*4e*
10			*3c*		*3c*
11				*4f*	*4f*
12		*2b*			*2b*
⋮					

GIF Terminator

In order to provide a synchronization for the termination of a GIF image file, a GIF decoder will process the end of GIF mode when the character 0x3B hex or ';' is found after an image has been processed.

Multimedia Systems
Lecture – 11

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

Popular Image File Formats

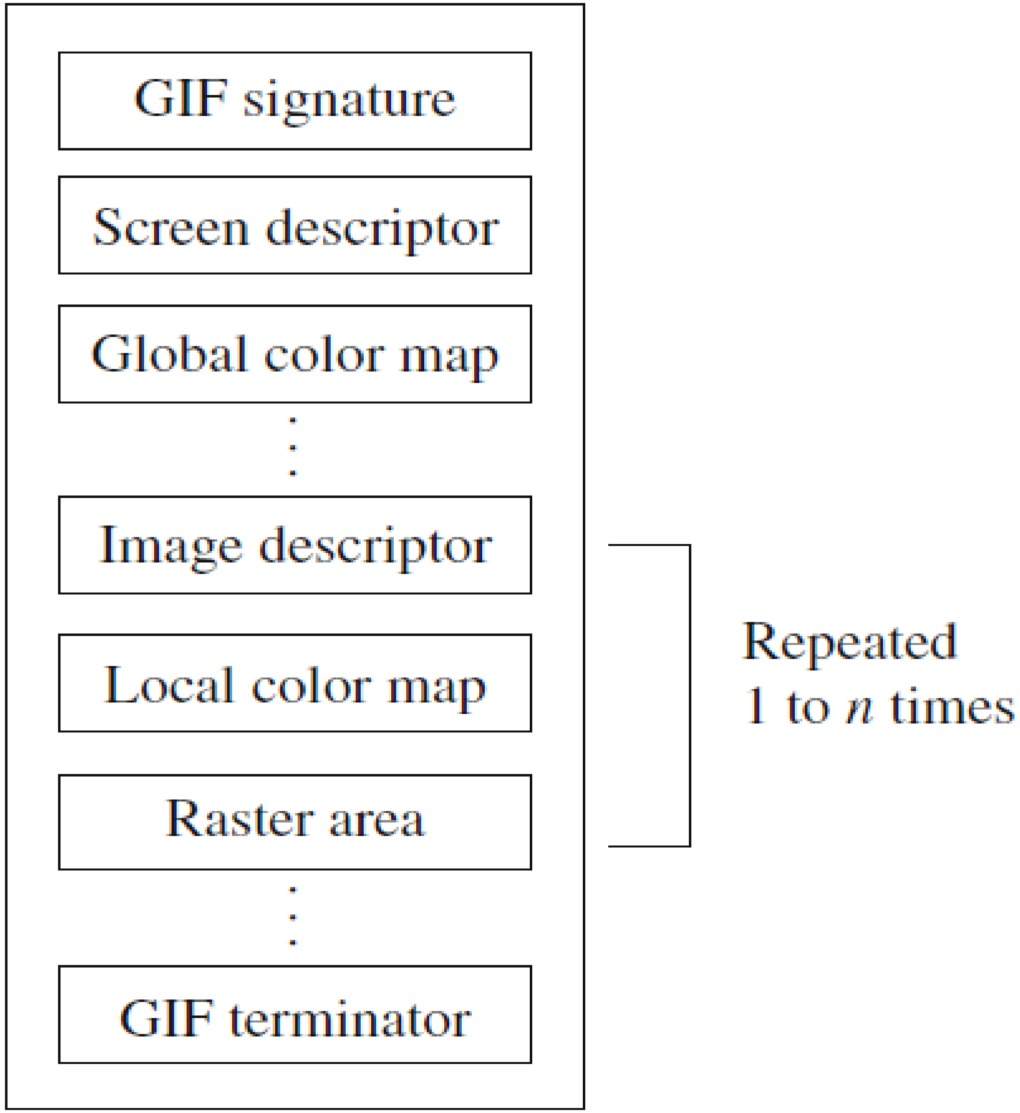
- 8-bit GIF: one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.
- JPEG: currently the most important common file format.
- PNG: most popular lossless image format.
- TIFF: flexible file format due to the addition of tags.
- EXIF: allows the addition of image metadata.
- PS and PDF: vector based language, popular in publishing and academia

GIF (Graphic Interchange Format)

- We examine GIF standard because it is so simple! yet contains many common elements.
- Limited to *8-bit (256) color images* only, which, while producing acceptable color images, is best suited for images with few distinctive colors (e.g., graphics or drawing).
- GIF uses the *LZW* compression algorithm that does not degrade the image quality.
- GIF standard supports *interlacing* — successive display of pixels in widely-spaced rows by a 4-pass display process.
- GIF actually comes in two flavors:
    1. *GIF87a*: The original specification.
    2. *GIF89a*: The later version. Supports simple animation via a Graphics Control Extension block in the data, provides simple control over delay time, a transparency index, etc.

GIF87

- Many such formats bear a resemblance to it but have grown a good deal more complex than this “simple” standard.

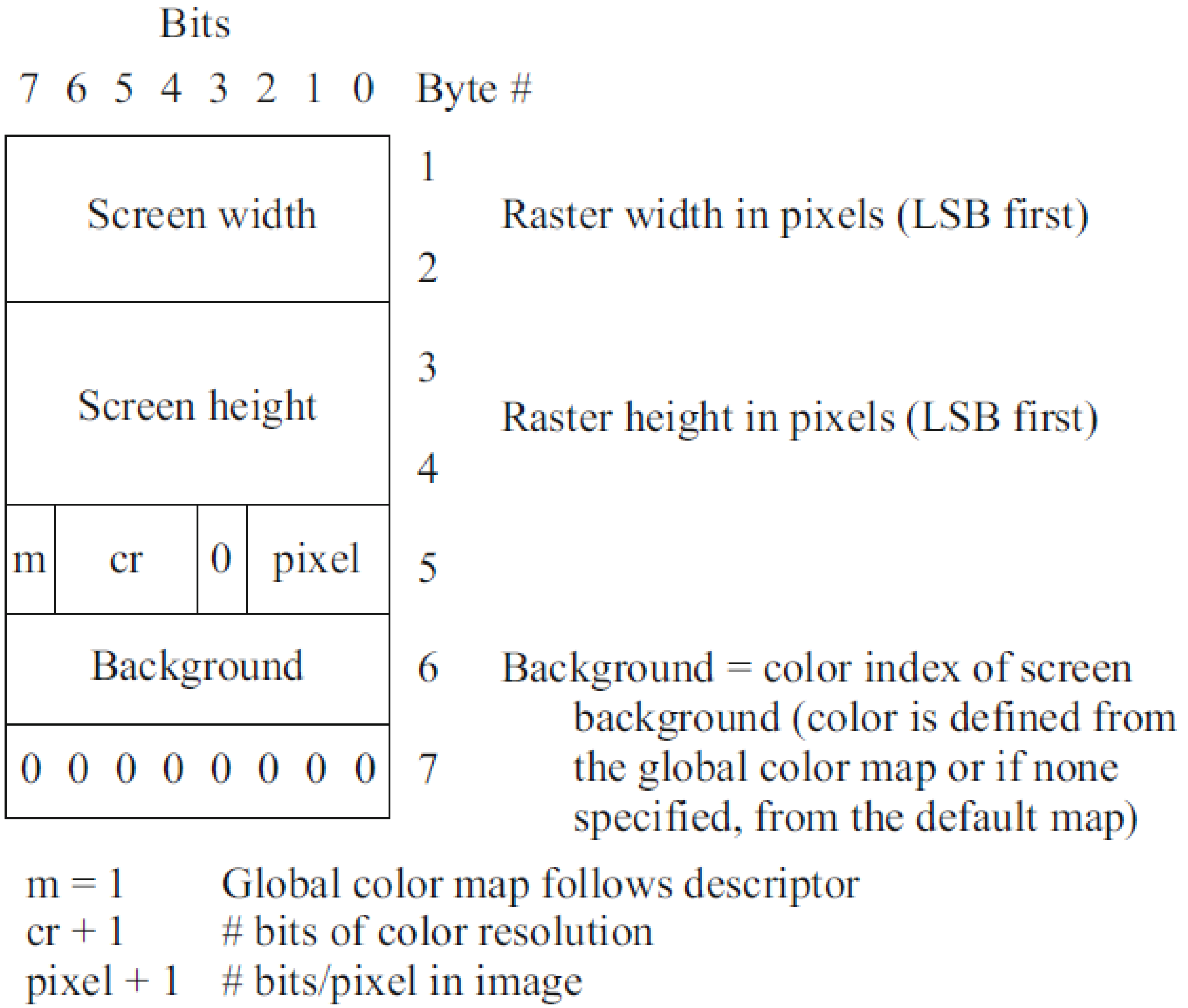


- The *Signature* is six bytes: GIF87a; the *Screen Descriptor* is a seven-byte set of flags.

- A GIF87 file can contain more than one image definition, usually to fit on several different parts of the screen.

- Therefore each image can contain its own color lookup table, a *Local Color Map*, for mapping 8 bits into 24-bit RGB values.

- The Screen Descriptor comprises a set of attributes that belong to every image in the file.



GIF Color Map

Bits 7 6 5 4 3 2 1 0	Byte #	
Red intensity	1	Red value for color index 0
Green intensity	2	Green value for color index 0
Blue intensity	3	Blue value for color index 0
Red intensity	4	Red value for color index 1
Green intensity	5	Green value for color index 1
Blue intensity	6	Blue value for color index 1
⋮		(continues for remaining colors)

GIF image descriptor

Bits 7 6 5 4 3 2 1 0	Byte #	
0 0 1 0 1 1 0 0	1	Image separator character (comma)
Image left	2 3	Start of image in pixels from the left side of the screen (LSB first)
Image top	4 5	Start of image in pixels from the top of the screen (LSB first)
Image width	6 7	Width of the image in pixels (LSB first)
Image height	8 9	Height of the image in pixels (LSB first)
m | i | 0 | 0 | 0 | pixel	10	m = 0 Use global color map, ignore ‘pixel’ m = 1 Local color map follows, use ‘pixel’ i = 0 Image formatted in Sequential order i = 1 Image formatted in Interlaced order pixel + 1 # bits per pixel for this image

Raster Area:

- The format of the actual image is defined as the series of pixel color index values that make up the image.
- The pixels are stored left to right sequentially for an image row.
- By default each image row is written sequentially, top to bottom.
- In the case that the Interlace or ‘i' bit is set in byte 10 of the Image Descriptor then the row order of the image display follows a four-pass process in which the image is filled in by widely spaced rows.
- The first pass writes every 8th row, starting with the top row of the image window. The second pass writes every 8th row starting at the fifth row from the top. The third pass writes every 4th row starting at the third row from the top. The fourth pass completes the image, writing every other row, starting at the second row from the top.

GIF four-pass interlace display row order

Image row	Pass 1	Pass 2	Pass 3	Pass 4	Result
0	*1a*				*1a*
1				*4a*	*4a*
2			*3a*		*3a*
3				*4b*	*4b*
4		*2a*			*2a*
5				*4c*	*4c*
6			*3b*		*3b*
7				*4d*	*4d*
8	*1b*				*1b*
9				*4e*	*4e*
10			*3c*		*3c*
11				*4f*	*4f*
12		*2b*			*2b*
⋮					

GIF Terminator

In order to provide a synchronization for the termination of a GIF image file, a GIF decoder will process the end of GIF mode when the character 0x3B hex or ';' is found after an image has been processed.

- To see how the file header looks, we can use command in the Unix/Linux/MacOS operating system: od (octal dump).
- In Unix, then, we issue the command

  od -c forestfire.gif | head -2

- we see the first 32 bytes interpreted as characters

```
G   I   F   8   7   a  \208 \2   \188 \1  \247 \0  \0  \6  \3 \5
J \132 \24 |    )  \7  \198 \195 \   \128 U   \27 \196 \166 & T
```

- To decipher the remainder of the file header (after GIF87a), we use hexadecimal:

  od -x forestfire.gif | head -2

- with the result

```
4749 4638 3761 d002 bc01 f700 0006 0305
ae84 187c 2907 c6c3 5c80 551b c4a6 2654
```

Multimedia Systems
Lecture – 12

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

JPEG

- The most important current standard for image compression is JPEG.
- This standard was created by a working group of the International Organization for Standardization (ISO) that was informally called the ***Joint Photographic Experts Group*** and is therefore so named.
- The human vision system has some specific limitations, which JPEG takes advantage of to achieve high rates of compression.
- The eye–brain system cannot see extremely fine detail.
- If many changes occur within a few pixels, we refer to that image segment as having *high* ***spatial frequency*** —that is, a great deal of change in (x, y) space.

- Therefore, color information in JPEG is decimated and then small blocks of an image are represented in the spatial frequency domain (u, v), rather than in (x, y).
- That is, the speed of changes in x and y is evaluated, from low to high, and a new "image" is formed by grouping the coefficients or weights of these speeds.
- Weights that correspond to slow changes are then favored, using a simple trick.
- Since we effectively throw away a lot of information by the division and truncation step, this compression scheme is "*lossy*"
- JPEG allows the user to set a desired *level of quality*, or *compression ratio* (input divided by output).

JPEG image with low quality specified by user.

This image is having a quality factor Q = 10. (The usual default quality factor is Q = 75). This image is a mere 15% of the original size. In comparison, a JPEG image with Q = 75 yields an image size 5.6% of the original, whereas a GIF version of this image compresses down to 23.0% of the uncompressed image size.

PNG

- **PNG format**: standing for ***Portable Network Graphics*** — system-independent image formats, meant to supersede the GIF standard, and extends it in important ways.
- Special features of PNG files include:
  - Support for up to ***16 bits*** per pixel in each color channel, i.e., 48-bit color.
  - Files may contain ***gamma-correction*** information for correct display of color images, as well as ***alpha-channel*** information for such uses as control of transparency.
  - The display progressively displays pixels in a 2-dimensional fashion by showing a few pixels at a time over ***seven*** passes through each ***8 x 8*** block of an image.
  - It supports both ***lossless*** and ***lossy*** compression with performance better than GIF.
  - PNG is widely supported by various web browsers and imaging software.

TIFF

- **TIFF**: stands for *Tagged Image File Format* is another popular image file format.
- The support for attachment of additional information (referred to as "tags") provides a great deal of flexibility.
- The most important tag is a ***format signifier***: what type of compression etc. is in use in the stored image.
- TIFF can store many different types of image: ***1-bit, grayscale, 8-bit color, 24-bit RGB***, etc.
- TIFF was originally a lossless format but now a JPEG tag allows one to opt for JPEG compression.

EXIF

- **EXIF** (*Exchange Image File*) is an image format for digital cameras.
- It enables the recording of image metadata (exposure, light source/flash, white balance, type of scene, etc.) for the standardization of image exchange.
- A variety of tags (many more than in TIFF) is available to facilitate higher quality printing, since information about the camera and picture-taking conditions can be stored and used, e.g., by printers for possible color-correction algorithms.

Windows BMP

- *BitMap* (BMP) is one major system standard image file format for Microsoft Windows.
- BMP supports many pixel formats, including indexed color (up to 8 bits per pixel), and 16, 24, and 32-bit color images.
- It makes use of ***Run-Length Encoding (RLE)*** compression.

Some Other Image Formats

- **Microsoft Windows WMF** (*Windows MetaFile*): the native vector file format for the Microsoft Windows operating environment:
- Consist of a collection of ***GDI (Graphics Device Interface)*** function calls, also native to the Windows environment.
- When a WMF file is "played" (typically using the Windows PlayMetaFile() function) the described graphics is rendered.
- WMF files are ***device-independent*** and are ***unlimited in size***.
- **Netpbm Format**: PPM (Portable PixMap), PGM (Portable GrayMap), and PBM (Portable BitMap) belong to a family of open source Netpbm formats. These formats are mostly common in the ***linux/unix environments***.

Multimedia Systems
Lecture – 13

By

Dr. Priyambada Subudhi

Assistant Professor

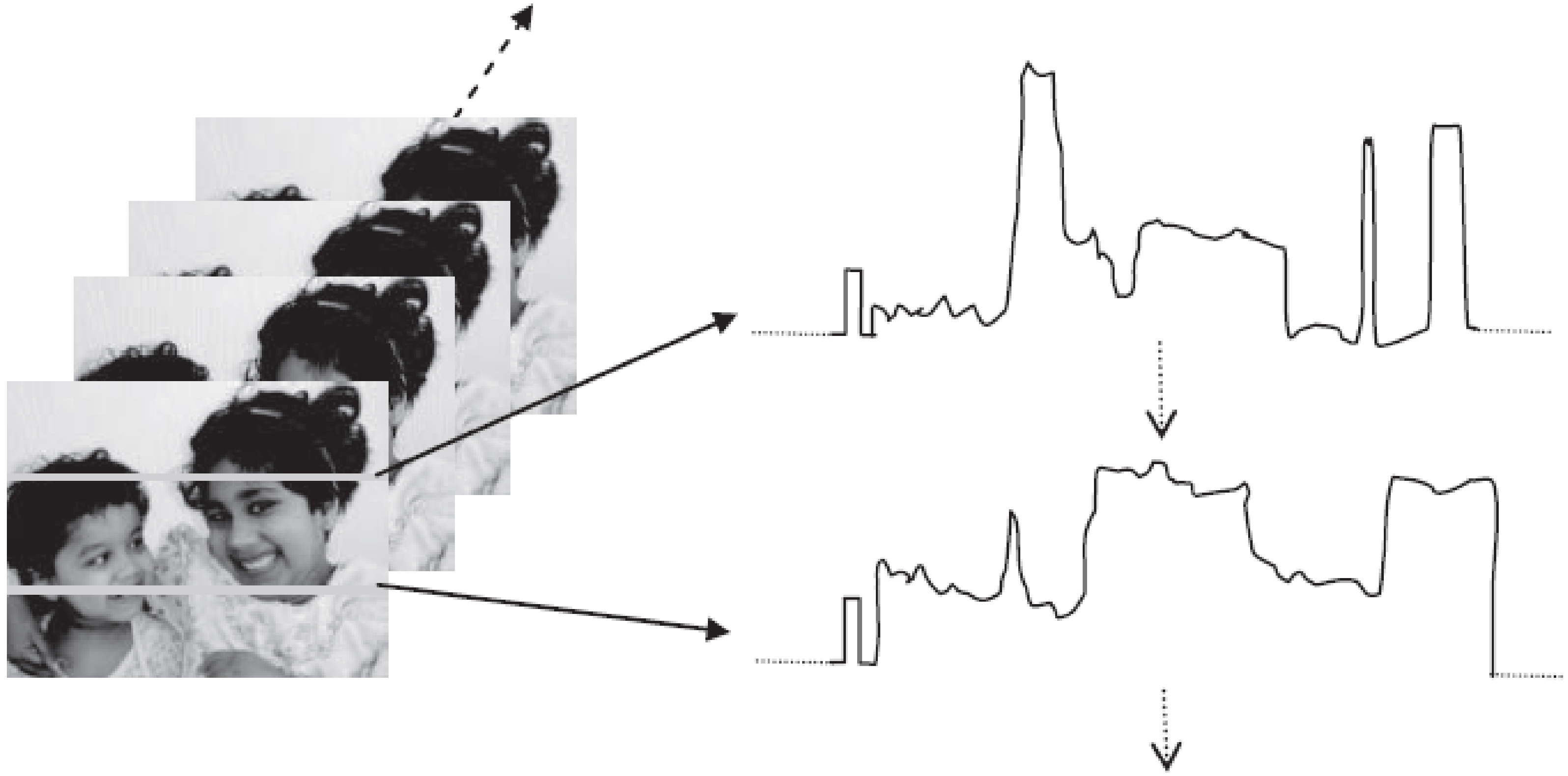
IIIT Sri City

Video

- Video, whether analog or digital, is represented by a sequence of discrete images shown in quick succession. Each image in the video is called a frame, which is represented as a matrix of pixels defined by a width, height, and pixel depth.
- In addition, two important properties govern video representation: frame rate and scanning format.
- The rate at which the images are shown is the frame rate.
- If the frame rate is too slow, the human eye perceives an unevenness of motion called flicker.

- Although digital video can be considered a three-dimensional signal—a 2D image changing over time—analog video is converted to a 1D signal of scan lines.
- This scan line conversion was introduced to make analog television broadcast technology work, and is central to the manner in which televisions (and all other cathode-ray tubes) display images.
- The electron gun(s) in a television project electrons on the phosphor screen from left to right in a scan line manner and from top to bottom successively for each frame.
- The phosphor screen glows at each location on a scan line creating a color at all positions on the line.
- Scanning formats, which is an outcome of the analog technology, can be represented as interlaced or progressive.

- *Left: Video is represented as a sequence of images. Right: Analog video of one frame scanned as a 1D signal. Each scan line is scanned from left to right as an analog signal separated by horizontal syncs. Two scan lines are shown; each begins with a horizontal sync and traces through the intensity variation on that scan line.*

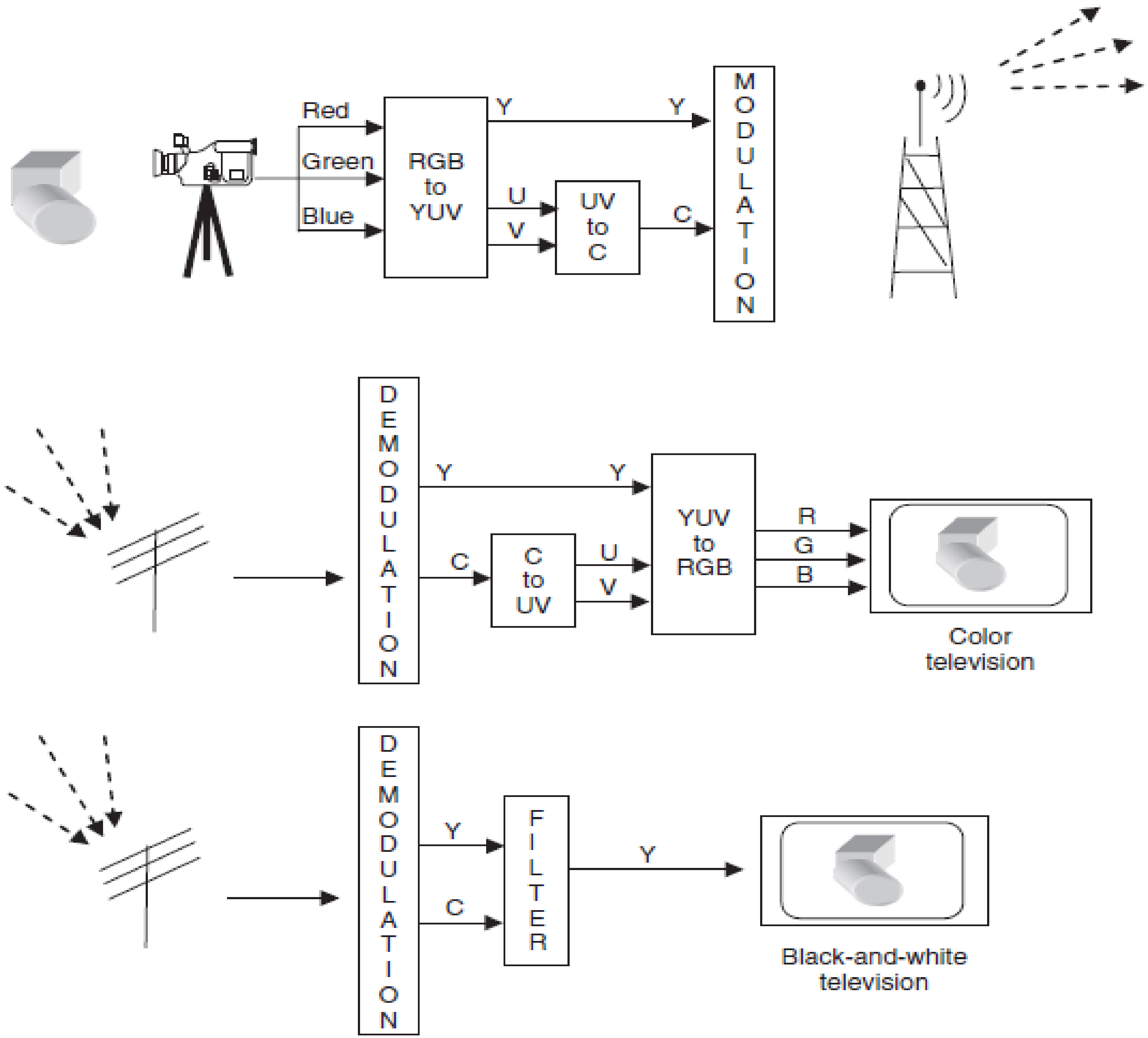


Digital display technologies display media in a digital format. Digital video display on these devices, such as LCD or plasma, does not require the scanning mechanism described previously. However, when the technology for digital video started to evolve, the television instruments were still rendering analog signals only. As a result, the digital video standards have their representations and formats closely tied to analog TV standards.

Analog Video and Television

- Analog video signal used in broadcast is scanned as a 1D signal in time.
- This 1D signal captures the time-varying image intensity information only along scanned lines.
- Television requires this analog scanned information to be broadcast from a broadcast station to all users.
- The standardization process implemented in the broadcast of analog video for television mandated a few requirements, which were necessary for making television transmission viable: ***YUV color space conversion*** and ***interlaced scanning***.

Television works by sending scan line information in interlaced YUV format.



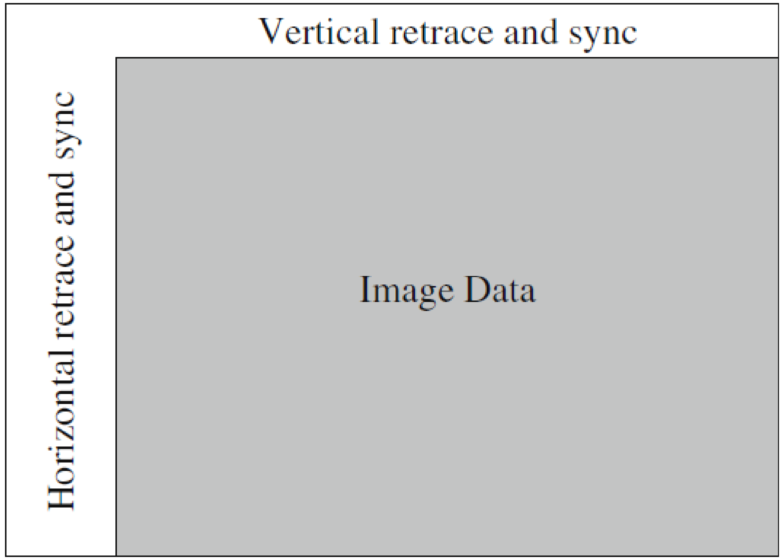
Conversion to YUV

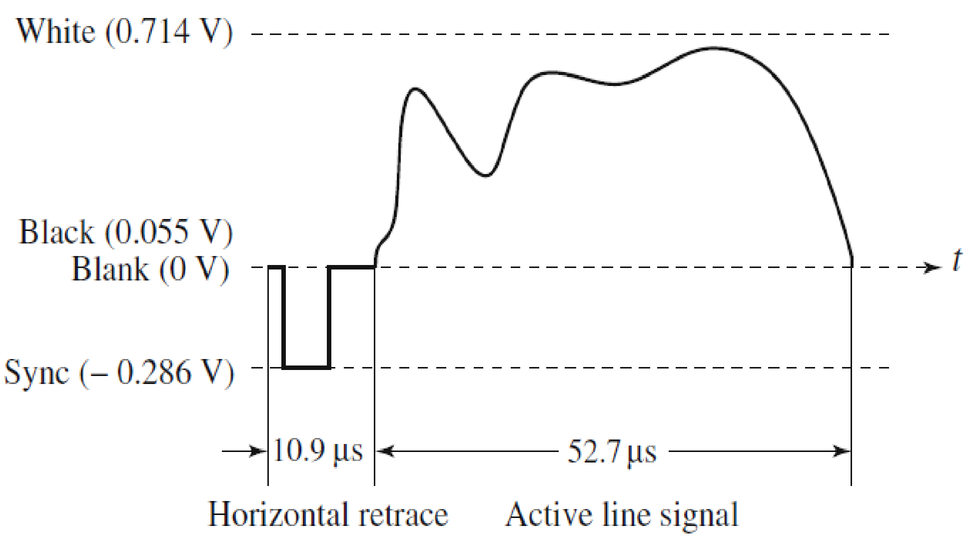
- Video frames, like images, are represented using a color format, which is normally RGB.
- This RGB color space is used by cathode-ray tube–based display devices, such as the television, to display and render the video signal.
- For transmission purposes, however, the RGB signal is transformed into a YUV signal.
- The YUV color space aims to decouple the intensity information (Y or luminance) from the color information (UV or chrominance).
- The separation was intended to reduce the transmission bandwidth and is based on experiments with the human visual system, which suggests that humans are more tolerant to color distortions than to intensity distortions.

Analog Video Scanning

- Video is scanned as a 1D signal, where each raster line is interspaced with horizontal and vertical syncs.
- For ***horizontal synchronization*** in analog video, a small voltage offset from zero is used to indicate black and another value, such as zero, to indicate the start of a line.
- ***Vertical synchronization*** is carried out by the cycles in the power outlet (60 Hz for NTSC, 50 Hz for PAL). Every 1/60th of a second, the electron gun is reset by the vertical sync to draw the beginning of the next frame.

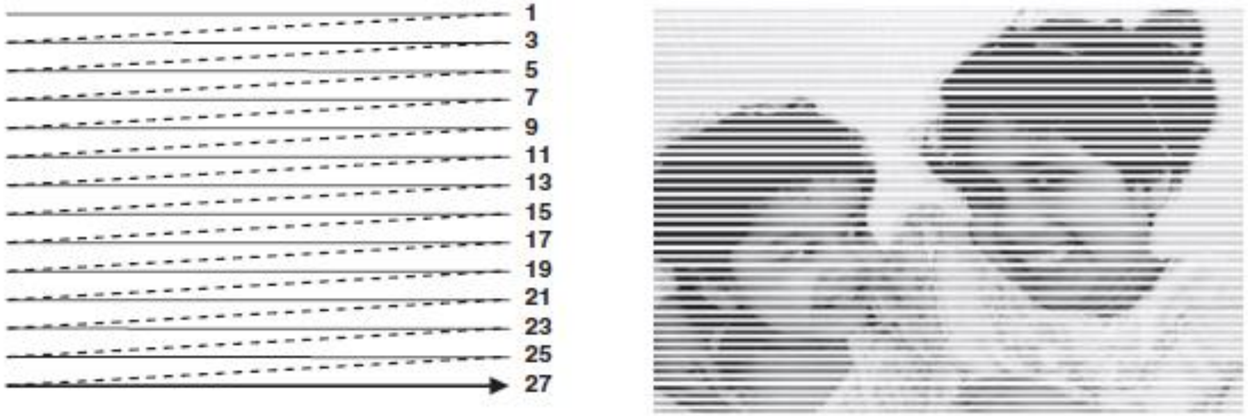
Horizontal Synchronization



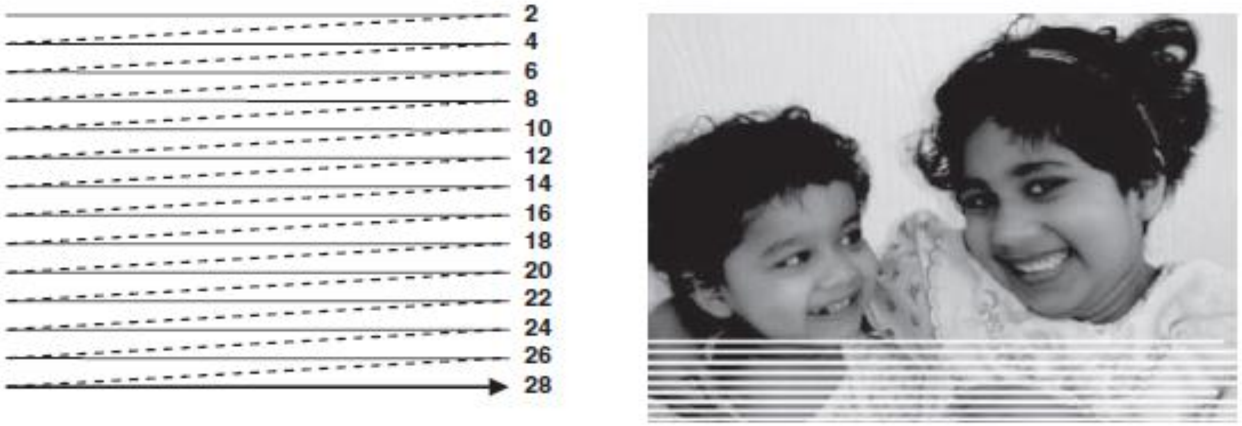


- In TV and in some monitors and multimedia standards, another system, *interlaced* scanning, is used.
- Here, the odd-numbered lines are traced first, then the even-numbered lines. This results in "odd" and "even" *fields*—two fields make up one frame.
- But the resulting video drawn by interlaced scanning techniques might be unacceptable and has occasional flicker and artifacts. This is caused because the video is captured at different moments in time as two field and, hence, interlaced video frames exhibit motion artifacts when both fields are combined and displayed at the same moment.
- Video is of better quality when it is captured progressively and drawn progressively, which eliminates the occasional flicker.

- *Interlaced scanning. The top figure shows the upper “odd” field consisting of odd-numbered lines. The bottom shows a lower “even” field interspersed with the odd field. Both fields are shown in succession to meet the required frame rate.*

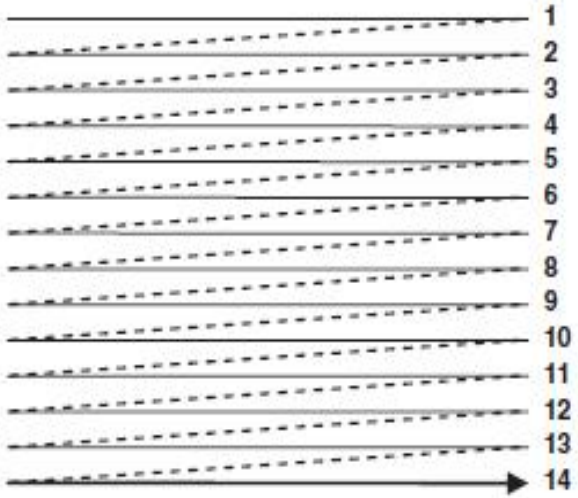


Upper field



Lower field

Progressive scanning. All the scan lines are drawn in succession, unlike in the interlaced case.



Types of Video Signals

Composite Video

- Composite video is also called ***baseband video*** or RCA video.
- Composite video contains both *chrominance (color) and luminance (brightness) information, along with synchronization and blanking pulses, all together in a single signal*.
- This was done to reduce bandwidth and achieve real-time transmission.
- However, in composite video, interference between the chrominance and luminance information is inevitable and tends to worsen when the signal is weak.
- This is why fluctuating colors, false colors, and intensity variations are seen when a distant NTSC television station sends signals that are weak and not properly captured at home with old-fashioned "rabbit ears," or outdoor "aerial" antennae.

S-Video

- S-Video (***Super-Video***, sometimes referred to as *Y/C Video*) is a video signal transmission in which the luminance signal and the chrominance signal are transmitted separately to achieve superior picture clarity.
- The luminance signal (*Y*) carries brightness information, and the chrominance signal (*C*) carries color information.
- Here, the chrominance signal (*C*) is formed by combining the two chrominance signals *U* and V into one signal along with their respective synchronization data, so at display time, the *C* signal can be separated into *U* and *V* signals.
- Separating the *Y* and *C* channels and sending them separately reduces problems caused by interference between the luminance and chrominance signals and yields a superior visual quality.

Component Video

- Component video strives to go a step further than S-Video by keeping all three *Y, U, V* (or equivalent) components separate.
- Consequently, the bandwidth required to broadcast component video is more than the composite or S-Video and, correspondingly, so is the visual quality.
- The separation of these components prevents artifacts due to intersignal interference.

Connectors for typical analog display interfaces. From left to right:, Composite video, S-video, and Component video

Multimedia Systems
Lecture – 14

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

Analog Video Standards

NTSC Video

- ***NTSC***, named for the ***National Television System Committee***, is the analog television system that is mostly used in most of North America and Japan.
- It uses a familiar ***4:3*** *aspect ratio* (i.e., the ratio of picture width to height) and ***525*** scan lines per frame at ***30 fps, interlaced*** fileds.
- NTSC uses the YIQ color model.
- NTSC video is an analog signal with no fixed horizontal resolution. Therefore, we must decide how many times to sample the signal for display.
- Each sample corresponds to one pixel output. A *pixel clock* divides each horizontal line of video into samples. The higher the frequency of the pixel clock, the more samples per line.

PAL Video

- ***PAL (Phase Alternating Line)*** is a TV standard originally invented by German scientists, is widely used in Western Europe, China, India etc.
- It uses ***625 scan lines*** per frame, at ***25 frames per second*** (or 40 msec / frame), with a ***4 : 3*** aspect ratio and ***interlaced*** fields.
- Its broadcast TV signals are also used in composite video.
- PAL uses the YUV color model with an 8 MHz channel, allocating a bandwidth of 5.5 MHz to Y and 1.8 MHz each to U and V.
- To improve picture quality, chroma signals have alternate signs (e.g., +U and — U) in successive scan lines; hence the name "***Phase Alternating Line***".
- This facilitates the use of a ***comb filter*** at the receiver — the signals in consecutive lines are averaged so as to cancel the chroma signals for separating Y and C and obtain high - quality Y signals.

SECAM Video

- SECAM, which was invented by the French, is the third major broadcast TV standard.
- SECAM stands for ***Systeme Electronique Couleur Avec Memorie***.
- SECAM also uses ***625 scan lines*** per frame, at ***25 frames per second***, with a ***4:3*** aspect ratio and ***interlaced fields***.
- SECAM and PAL are similar, differing slightly in their color coding scheme.
- In SECAM, U and V signals are modulated using separate color subcarriers at 4.25 MHz and 4.41 MHz, respectively. They are sent in alternate lines - that is, only one of the U or V signals will be sent on each scan line.

Comparison of analog broadcast TV systems

TV system	Frame rate (fps)	Number of scan lines	Total channel width (MHz)	Bandwidth allocation (MHz)		
				Y	I or U	Q or V
NTSC	29.97	525	6.0	4.2	1.6	0.6
PAL	25	625	8.0	5.5	1.8	1.8
SECAM	25	625	8.0	6.0	2.0	2.0

Digital Video

The advantages of digital representation for video are many. It permits

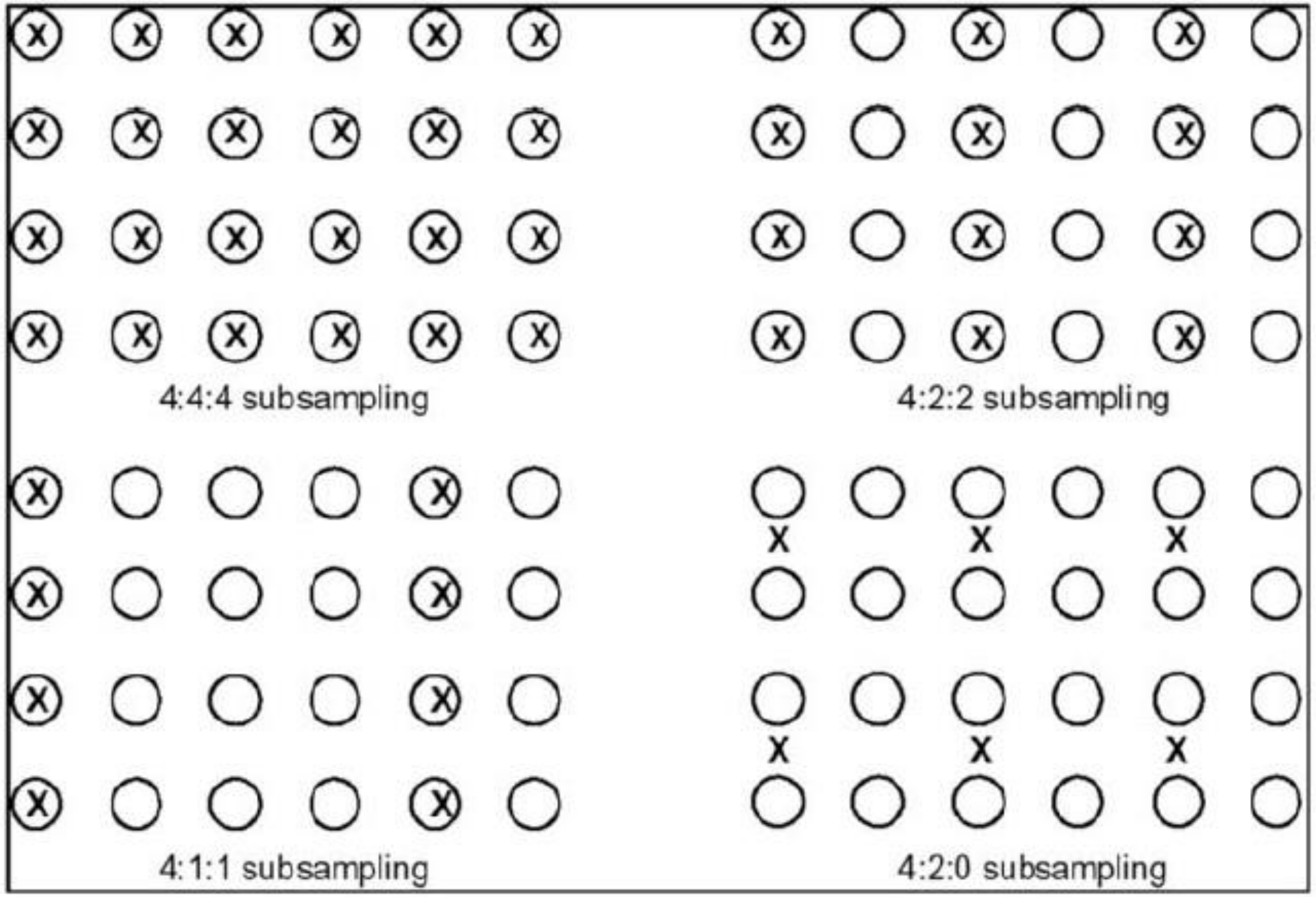
- Storing video on digital devices or in memory, ready to be processed (noise removal, cut and paste, and so on) and integrated into various multimedia applications.
- Direct access, which makes nonlinear video editing simple.
- Repeated recording without degradation of image quality.
- Ease of encryption and better tolerance to channel noise.

YUV Subsampling /Chroma Subsampling

- Video signals captured by digital cameras are represented in the RGB color space, which is also used to render video frames on a display device.
- However, for transmission and other intermediary processing, the YUV space is commonly used.
- The YUV space separates the color and luminance information.
- The color information (UV) is then further subsampled to gain more bandwidth.
- In analog video, subsampling is achieved by allocating half as much bandwidth to chrominance as to luminance.
- In digital video, subsampling can be done by ***reducing the number of bits used for the color channels on average***.

- Depending on the way subsampling is done, a variety of subsampling ratios can be achieved.
- The circles represent pixel information.
- Potentially, we could store 1 byte each for Y, U, and V components, resulting in ***24 bits per pixel***.
- In subsampling, the luminance component Y is left untouched—that is, 1 byte is reserved for the luminance data per pixel.
- An X at a pixel position suggests that we also store the chrominance components for this position.

YUV subsampling schemes used in video



- In the ***4:4:4*** scheme, each pixel has luminance (8 bits) and chrominance (8 bits for U and 8 bits for V), resulting in ***24 bits per pixel***.
- In the ***4:2:2*** subsampling scheme, chrominance information is stored for every other pixel bringing the equivalent bits per pixel down to ***16***.
- In the ***4:1:1*** subsampling scheme, chrominance is stored every fourth pixel in a row.
- Whereas in the ***4:2:0*** scheme, the average of the U values for a 2×2 pixel area is stored, and similarly for the V values.
- Since there is only 1 U and 1 V sample for every four luminance samples, the equivalent bits per pixel is brought down to ***12 bits per pixel***.

Multimedia Systems
Lecture – 14

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

CCIR and ITU-R Standards for Digital Video

- The CCIR is the *Consultative Committee for International Radio*. One of the most important standards it has produced is ***CCIR-601*** for component digital video.
- This standard has since become standard ***ITU-R Rec. 601***, an international standard for professional video applications.
- It is adopted by several digital video formats, including the popular DV video.
- The NTSC version has 525 scan lines, each having 858 pixels. Because the NTSC version uses 4:2:2, each pixel can be represented with two bytes (8 bits for *Y* and 8 bits alternating between *U* and *V*). The Rec. 601 (NTSC) data rate is thus approximately 216Mbps.

- The ***CIF*** format (Common Interchange Format) was established for a ***progressive digital broadcast television***.
- It consists of VHS quality resolutions whose width and height are divisible by 8—a requirement for digital encoding algorithms.
- The Quarter Common Interchange Format (***QCIF***) was established for ***digital videoconferencing over ISDN lines***.
- CIF is a compromise between ***NTSC and PAL***, in that it adopts the ***NTSC frame rate*** and ***half the number of active lines in PAL***.
- When played on existing TV sets, NTSC TV will first need to convert the number of lines, whereas PAL TV will require frame rate conversion.

ITU-R digital video specifications

	Rec. 601 525/60 NTSC	Rec. 601 625/50 PAL/SECAM	CIF	QCIF
Luminance resolution	720 × 480	720 × 576	352 × 288	176 × 144
Chrominance resolution	360 × 480	360 × 576	176 × 144	88 × 72
Color subsampling	4:2:2	4:2:2	4:2:0	4:2:0
Aspect ratio	4:3	4:3	4:3	4:3
Fields/sec	60	50	30	30
Interlaced	Yes	Yes	No	No

High-Definition TV (HDTV)

- The usual NTSC analog TV signal in the United States has 525 scan lines, with 480 actually visible.
- The usual TV has an effective picture resolution of about 210,000 pixels.
- Today, consumers are accustomed to better resolutions such as 1024×768 and even higher, which are now commonly supported by most graphics hardware that come with computers.
- A class of digital television called HDTV supports a higher resolution display format along with surround sound.
- The visual formats used in HDTV are as follows:
  - 720*p*—1280×720 pixels progressive
  - 1080*i*—1920×1080 pixels interlaced
  - 1080*p*—1920×1080 pixels progressive

- They use the MPEG2-based video compression format with a 17 Mbps bandwidth.
- Although HDTV signals can be stored and transmitted effectively using MPEG-2 technology, a lot of bandwidth is required to transmit numerous channels.
- The aspect ratio of HDTV is 16:9 (1.78:1
- .The increased resolution provides for a clearer, more detailed picture.
- In addition, progressive scan and higher frame rates result in a picture with less flicker and better rendering of fast motion.

Ultra High Definition TV (UHDTV)

- UHDTV is a new development—a new generation of HDTV.
- The standards announced in 2012 support 4K UHDTV: 2160P (3,840×2,160, progressive scan) and 8K UHDTV: 4320P (7,680 × 4,320, progressive scan).
- The aspect ratio is 16:9. The bit-depth can be up to 12 bits, and the chroma subsampling can be 4:2:0 or 4:2:2.
- The supported frame rate has been gradually increased to 120 fps.
- The UHDTV will provide superior picture quality, but it will require a much higher bandwidth and/or bitrate.

Digital Display Interfaces

- Given the rise of digital video processing and the monitors that directly accept digital video signals, there is a great demand toward video display interfaces that transmit digital video signals.

- The most widely used digital video interfaces include
  - Digital Visual Interface (DVI)
  - High- Definition Multimedia Interface (HDMI), and
  - DisplayPort

Connectors of different digital display interfaces. From left to right: VGA (For analog video), DVI, HDMI, DisplayPort

Digital Visual Interface (DVI)

- Digital Visual Interface (DVI) was for transferring digital video signals, particularly from a ***computer's video card to a monitor***.
- It carries uncompressed digital video and can be configured to support multiple modes, including ***DVI-D (digital only), DVI-A (analog only), or DVI-I (digital and analog).***
- The support for analog connections makes DVI backward compatible with VGA (Video Graphics Array).

- Through DVI, a source, e.g., video card, can read the display’s *extended display identification data* (EDID), which contains the display’s identification, color characteristics, and table of supported video modes.
- When a source and a display are connected, the source first queries the display’s capabilities by reading the monitor’s EDID block.
- A preferred mode or native resolution can then be chosen.
- The maximum pixel clock frequency of DVI is 165MHz, which supports a maximum resolution of 2.75megapixels at the 60Hz refresh rate.
- This allows a maximum 16:9 screen resolution of 1,920×1,080 at 60 Hz.

High-Definition Multimedia Interface (HDMI)

- HDMI is a newer digital audio/video interface developed to be backward-compatible with DVI.
- Its electrical specifications, in terms of TMDS and VESA/DDC links, are identical to those of DVI.
- HDMI, however, differs from DVI in the following aspects:
  - HDMI does not carry analog signal and hence is not compatible with VGA.
  - DVI is limited to the RGB color range (0–255). HDMI supports both RGB and YUV 4:4:4 or 4:2:2. The latter are more common in application fields other than computer graphics.
  - HDMI supports digital audio, in addition to digital video.
- The maximum pixel clock rate for HDMI 1.0 is 165MHz, HDMI 1.3 increases that to 340MHz while the latest HDMI 2.0 supports 4K resolution at 60 fps.

DisplayPort

- DisplayPort is the first display interface that uses ***packetized data transmission***, like the Internet or Ethernet.
- Specifically, it is based on small data packets known as ***micro packets***, which can embed the clock signal within the data stream.
- DisplayPort can achieve a higher resolution yet with fewer pins than the previous technologies.
- The use of data packets also allows DisplayPort to be extensible.
- DisplayPort can be used to transmit audio and video simultaneously, or either of them.
- It has a much higher video bandwidth, enough for four simultaneous 1080P 60Hz displays, or 4K video at 60 Hz.

- Compared with HDMI, DisplayPort has slightly more bandwidth, which also accommodates multiple streams of audio and video to separate devices.
- It is royalty-free, while HDMI charges an annual fee to manufacturers. These points make DisplayPort a strong competitor to HDMI in the consumer electronics market

Multimedia Systems
Lecture – 15

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

3D Video and TV

- Three-dimensional (3D) pictures and movies have been in existence for decades.
- Increasingly, it is in movie theaters, broadcast TV (e.g., sporting events), personal computers, and various handheld devices.
- The main advantage of the 3D video is that it enables the experience of immersion— be there, and really Be there.
- We will see fundamentals of 3D vision or 3D percept, emphasizing stereo vision (or stereopsis) since most modern 3D video and 3D TV are based on stereoscopic vision.

Cues for 3D Percept

- The human vision system is capable of achieving a 3D percept by utilizing multiple cues.
- They are combined to produce optimal depth estimates.
- When the multiple cues agree, this enhances the 3D percept.
- When they conflict with each other, the 3D percept can be hindered.

Monocular Cues

- The monocular cues that do not necessarily involve both eyes include:
  - *Shading*—depth perception by shading and highlights
  - *Perspective scaling*—converging parallel lines with distance and at infinity
  - *Relative size*—distant objects appear smaller compared to known same-size objects not in distance
  - *Texture gradient*—the appearance of textures change when they recede in distance
  - *Blur gradient*—objects appear sharper at the distance where the eyes are focused, whereas nearer and farther objects are gradually blurred
  - *Haze*—due to light scattering by the atmosphere, objects at distance have lower contrast and lower color saturation
  - *Occlusion*—a far object occluded by nearer object(s)
  - *Motion parallax*—induced by object movement and head movement, such that nearer objects appear to move faster.
- Among the above monocular cues, it has been said that Occlusion and Motion parallax are more effective.

Binocular Cues

- The human vision system utilizes effective binocular vision, i.e., *stereo vision*.
- Our left and right eyes are separated by a small distance, on average approximately 2.5 inches, or 65mm. This is known as the *interocular distance*.
- As a result, the left and right eyes have slightly different views, i.e., images of objects are shifted horizontally.
- The amount of the shift, or *disparity*, is dependent on the object's distance from the eyes, i.e., its *depth*, thus providing the binocular cue for the 3D percept.
- The horizontal shift is also known as *horizontal parallax*.
- The fusion of the left and right images into single vision occurs in the brain, producing the 3D percept.

3D Movie and TV Based on Stereo Vision

- **3D Movie Using Colored Glasses**
- **3D Movies Using Circularly Polarized Glasses**
- **3D TV with Shutter Glasses**

3D Movie Using Colored Glasses

- In the early days, most movie theaters offering a 3D experience provided glasses tinted with complementary colors, usually *red* on the left and *cyan* on the right. This technique is called *Anaglyph 3D*.
- Anaglyph 3D images contain two differently filtered colored images, one for each eye. The left image is filtered to remove Blue and Green, and the right image is filtered to remove Red.
- When viewed through the "color-coded" "anaglyph glasses", each of the two images reaches the eye it's intended for, revealing an integrated stereoscopic image.
- The visual cortex of the brain fuses this into the perception of a three-dimensional scene or composition.
- The Anaglyph 3D movies are easy to produce. However, due to the color filtering, the ***color quality is not necessarily the best***.

Anaglyph 3d glasses

An Anaglyph image

3D Movies Using Circularly Polarized Glasses

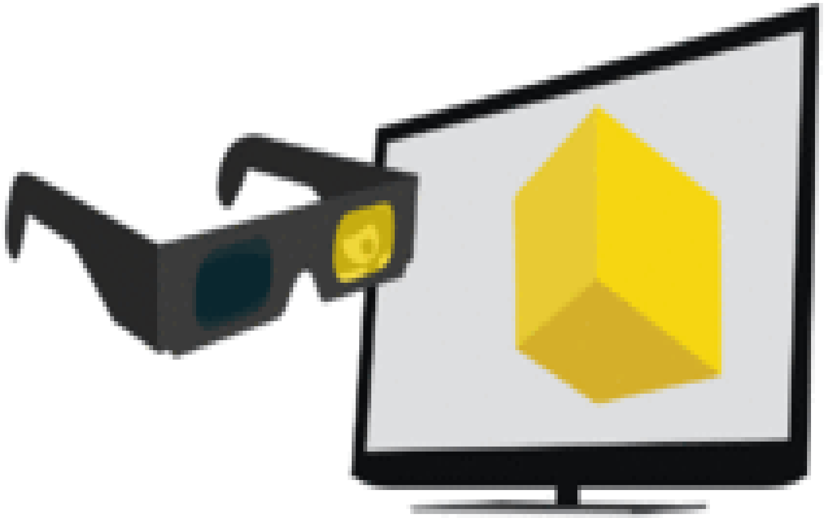
- Nowadays, the dominant technology in 3D movie theaters is the RealD Cinema System.
- Movie-goers are required to wear polarized glasses in order to see the movie in 3D.
- Basically, the lights from the left and right pictures are polarized in different directions. They are projected and superimposed on the same screen.
- The left and right polarized glasses that the audience wear are polarized accordingly, which allows one of the two polarized pictures to pass through while blocking the other.
- Circularly polarized glasses are used so the users can tilt their heads and look around a bit more freely without losing the 3D percept.

Polarized 3D systems

3D TV with Shutter Glasses

- Most TVs for home entertainment, however, use *Shutter Glasses*.
- Basically, the liquid crystal layer on the glasses that the user wears becomes opaque (behaving like a shutter) when some voltage is applied. It is otherwise transparent.
- The glasses are actively (e.g., via Infra-Red) synchronized with the TV set that alternately shows left and right images (e.g., 120Hz for the left and 120Hz for the Right) in a Time Sequential manner.
- 3D vision with shutter glasses can readily be realized on desktop computers or laptops with a modest addition of specially designed hardware and software. The NVIDIA GeForce 3D Vision Kit is such an example.

A pair of Crystal Eyes shutter glasses



Multimedia Systems
Lecture – 17

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

Audio

- Audio information is crucial for multimedia presentations and, in a sense, is the simplest type of multimedia data.

- However, some important differences between audio and image information cannot be ignored.

- For example, while it is customary and useful to occasionally drop a video frame from a video stream, to facilitate viewing speed, we simply cannot do the same with sound information or all sense will be lost from that dimension.

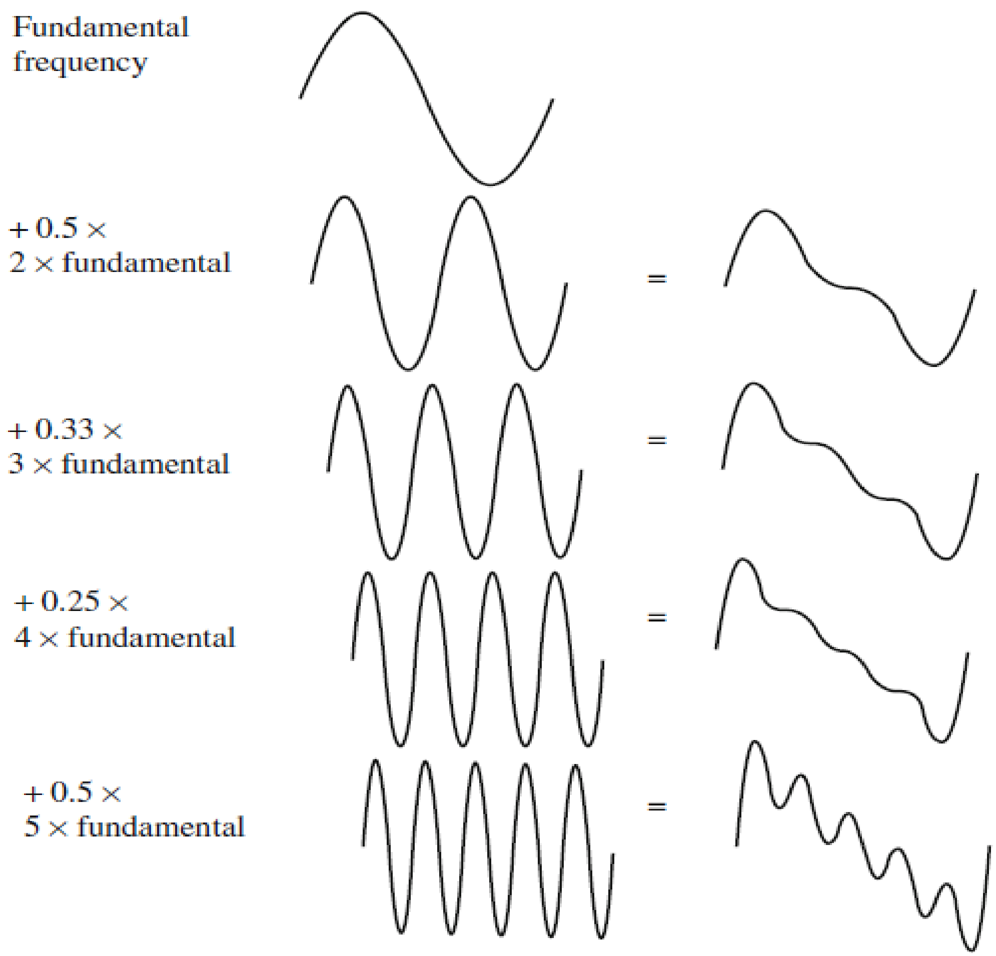
Digitization of Sound

What is Sound?

- Sound is a wave phenomenon like light, but it is macroscopic and involves molecules of air being compressed and expanded under the action of some physical device.
- For example, a speaker in an audio system vibrates back and forth and produces a longitudinal pressure wave that we perceive as sound.
- Without air there is no sound—for example, in space.
- Since sound is a pressure wave, it takes on continuous values, as opposed to digitized ones with a finite range.
- Nevertheless, if we wish to use a digital version of sound waves, we must form digitized representations of audio information.

- Although such pressure waves are longitudinal, they still have ordinary wave properties and behaviors, such as reflection (bouncing), refraction (change of angle when entering a medium with a different density), and diffraction (bending around an obstacle). This makes the design of “surround sound” possible.
- In general, any signal can be decomposed into a sum of sinusoids, if we are willing to use enough sinusoids.
- A weighted sinusoids can build up quite a complex signal.

Building up a complex signal by superposing sinusoids

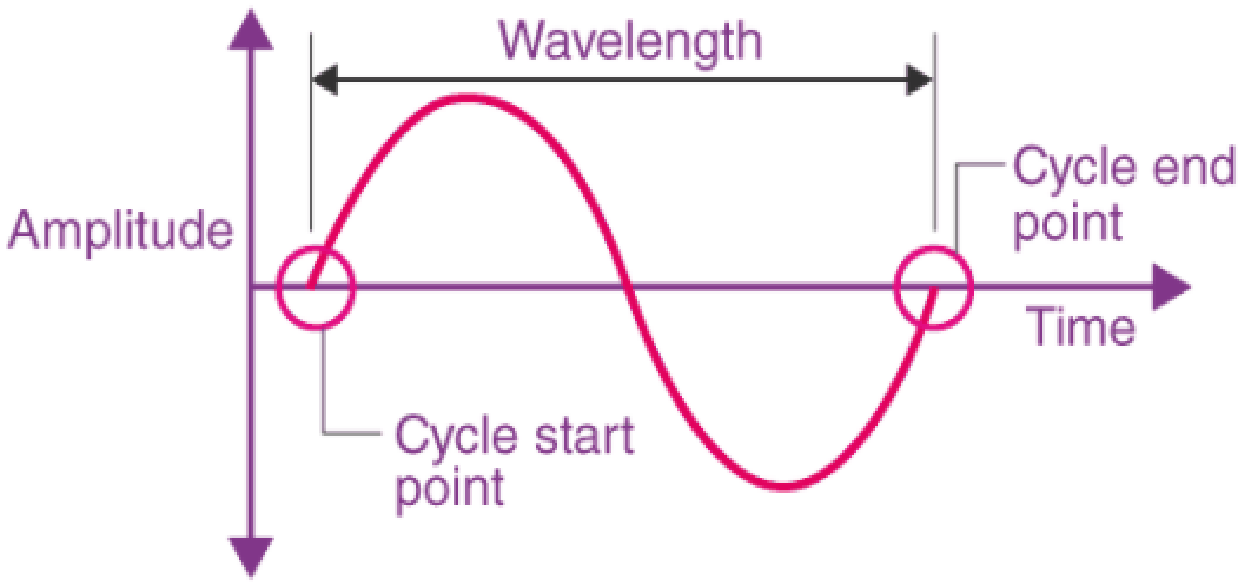


Characteristics of Sound

- The sound wave is having the following characteristics
- Amplitude:
  - It refers to the distance of the maximum vertical displacement of the wave from its mean position.
  - In sound, amplitude refers to the magnitude of compression and expansion experienced by the medium the sound wave is travelling through.
  - This amplitude is perceived by our ears as loudness. High amplitude is equivalent to loud sounds.
- Wavelength:
  - A sound wave is made of areas of high pressure alternated by an area of low pressure.
  - The high-pressure areas are represented as the peaks of the graph. The low-pressure areas are represented as troughs of the graph.
  - The physical distance between two consecutive peaks in a sound wave is referred to as the wavelength of the sound wave.

- Frequency/ Pitch of the Sound Waves
  - Frequency in a sound wave refers to the rate of the vibration of the sound travelling through the air. This parameter decides whether a sound is perceived as high pitched or low pitched.
  - In sound, the frequency is also known as **Pitch**.
  - The frequency of the vibrating source of sound is calculated in cycles per second (Hertz).

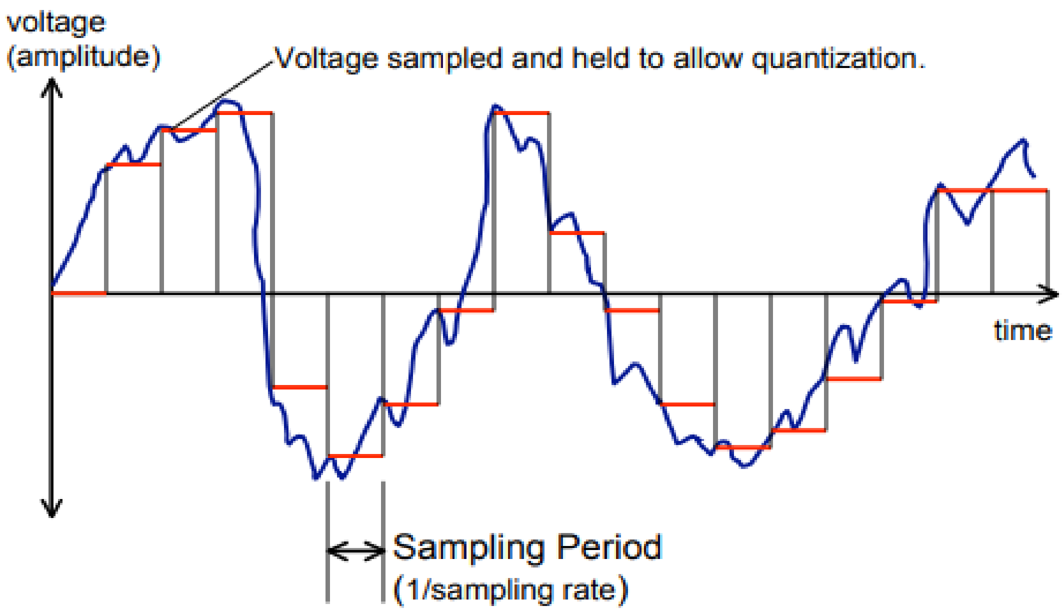
A depiction of Sound Waves in Waveform



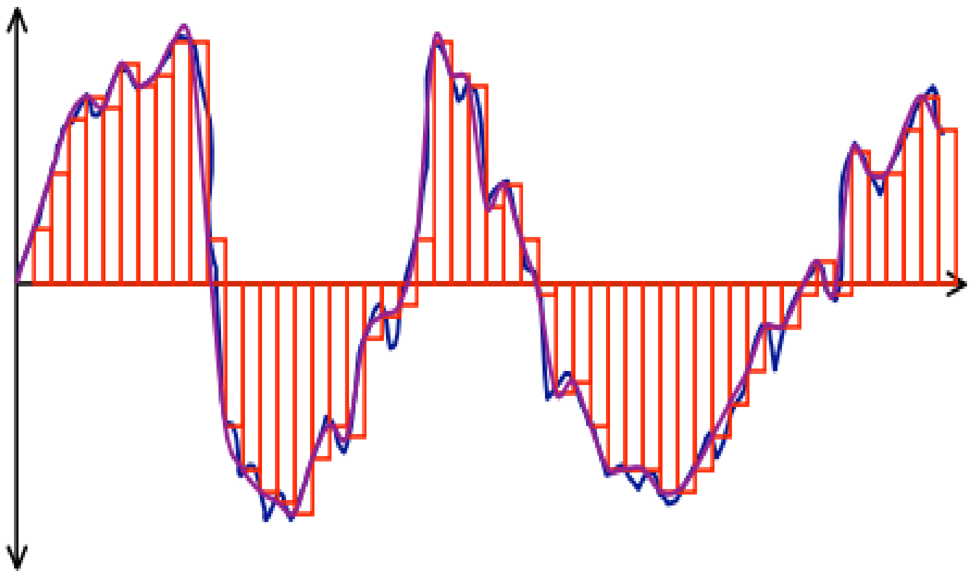
Digitization

- Since there is only one independent variable in sound i.e. time, we call this a **1D signal**.
- The amplitude value is a continuous quantity. Digitization means conversion to a stream of numbers.
- To fully digitize the sound signal, we have to ***sample in time and in amplitude.***
- ***Sampling*** means measuring the quantity we are interested in, usually at evenly spaced intervals.
- The first kind of sampling—using measurements only at evenly spaced *time* intervals—is simply called *sampling* and the rate at which it is performed is called the ***sampling rate*** or ***sampling frequency***.

Sampling

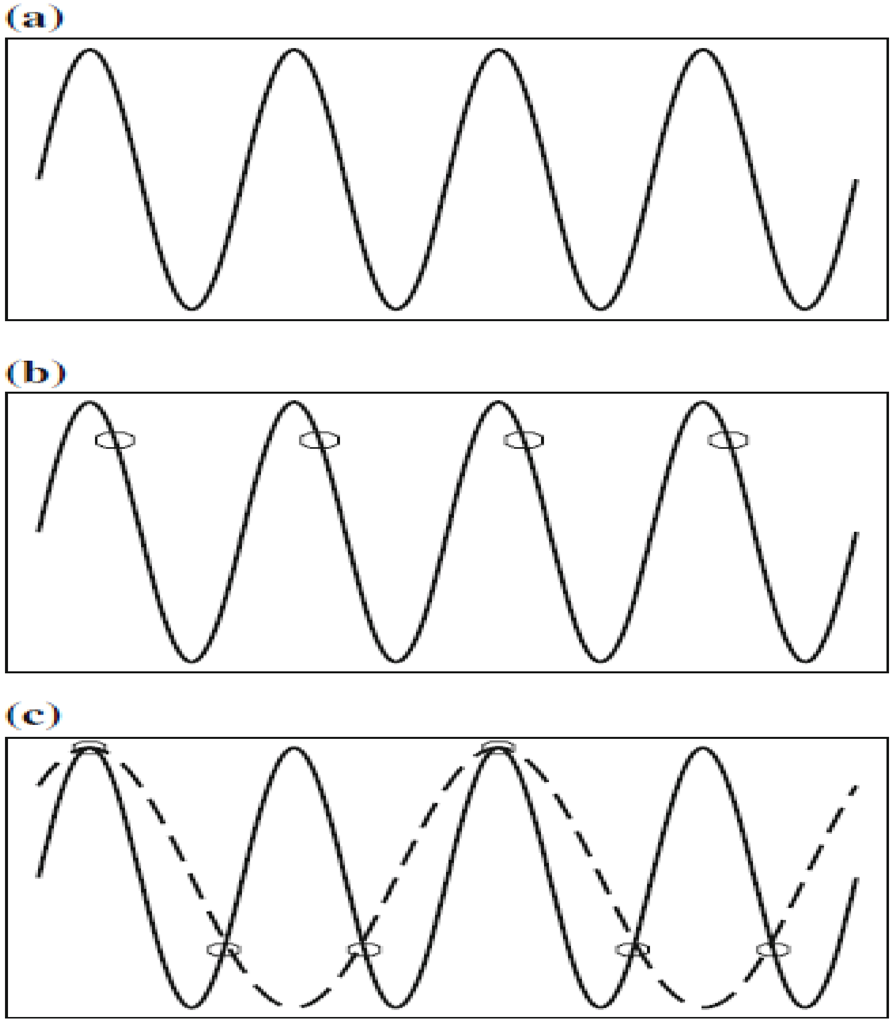


voltage
(amplitude)
Voltage sampled and held to allow quantization.
time
Sampling Period
(1/sampling rate)



- For audio, typical sampling rates are from ***8 kHz*** (8,000 samples per second) to ***48 kHz***.
- The human ear can hear from about *20 Hz to as much as 20 kHz*; above this level, we enter the range of ultrasound.
- The human voice can reach approximately 4 kHz.
- *Nyquist sampling rate :*
    - To preserve the full information in the signal, it is necessary to sample at twice the maximum frequency of the signal. This is known as the Nyquist rate.
    - If we sample the signal at a frequency that is lower that the Nyquist rate, when the signal is converted back into a continuous time signal, it will exhibit a phenomenon called ***aliasing***. Aliasing is the presence of unwanted components in the reconstructed signal.

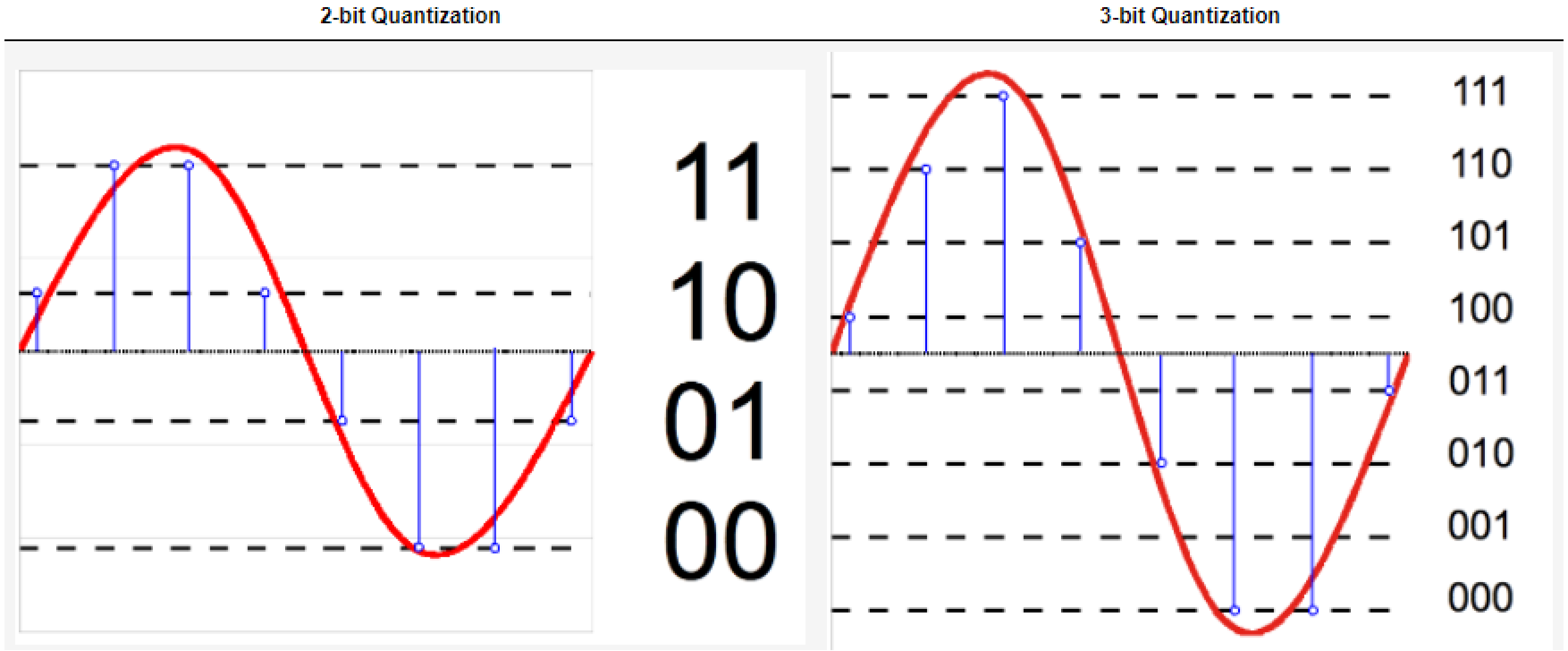
Aliasing: **a)** a single frequency; **b)** sampling at exactly the frequency produces a constant; **c)** sampling at 1.5 times per cycle produces an *alias* frequency that is perceived



Quantization:

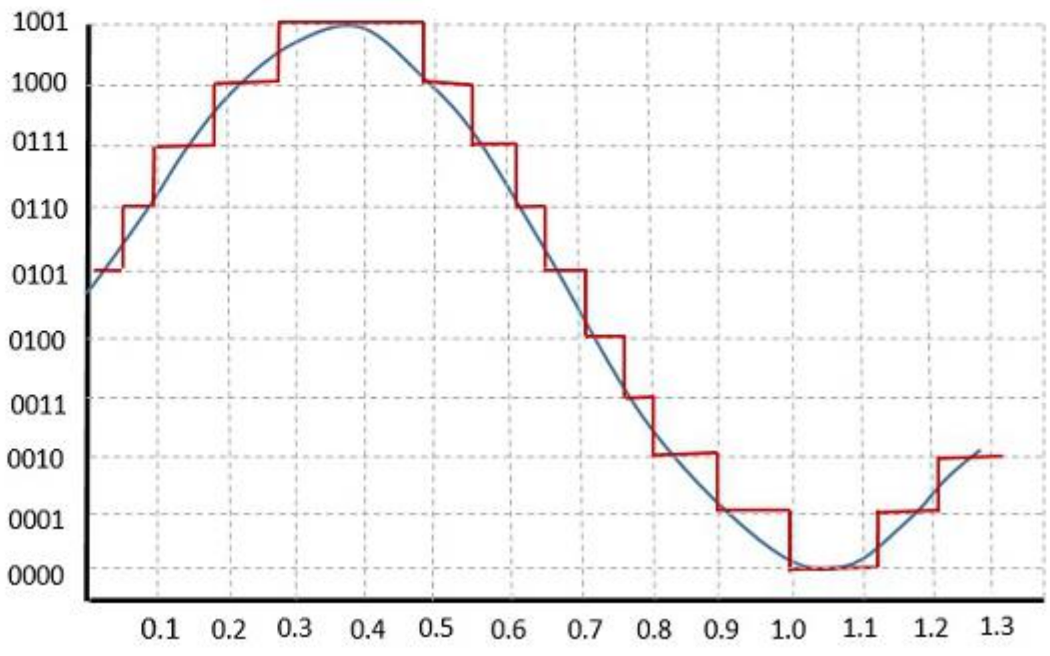
- Sampling in the amplitude dimension is called ***quantization*** *or It* refers to the process of transforming a sampled analog signal, to a digital signal, which has a discrete set of values.
- While we have discussed only uniform sampling, with equally spaced sampling intervals, non-uniform sampling is possible. This is not used for sampling in time but is used for quantization.
- Typical uniform quantization rates are 8-bit and 16-bit; 8-bit quantization divides the vertical axis into 256 levels, and 16-bit divides it into 65,536 levels.
- Quantization Error: The difference between an input value and its quantized value is called a **Quantization Error**.
- A digitized sample can have a maximum error of one-half the discretization step size.

2-bit and 3-bit Quantization



Linear and Nonlinear/ Uniform and Non-uniform Quantization

- Samples are typically stored as uniformly quantized values. This is called *linear or uniform quantization.*



- There are two types of uniform quantization. They are *Mid-Rise* type and *Mid-Tread* type.
- The **Mid-Rise** type is so called because the origin lies in the middle of a raising part of the stair-case like graph. The quantization levels in this type are even in number.
- The **Mid-tread** type is so called because the origin lies in the middle of a tread of the stair-case like graph. The quantization levels in this type are odd in number.

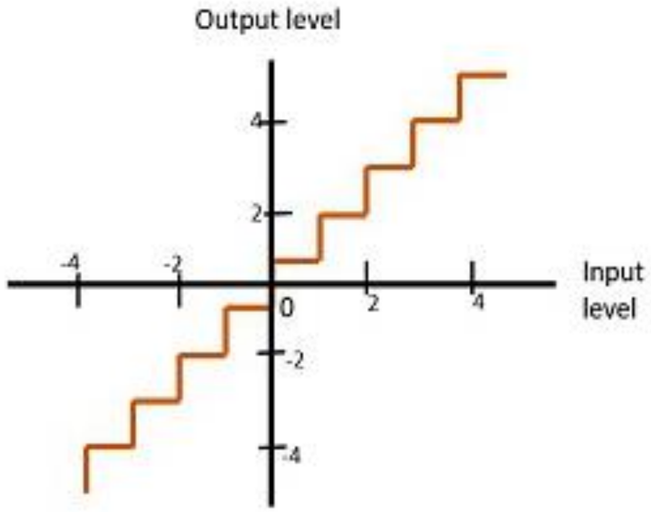


Fig 1 : Mid-Rise type Uniform Quantization

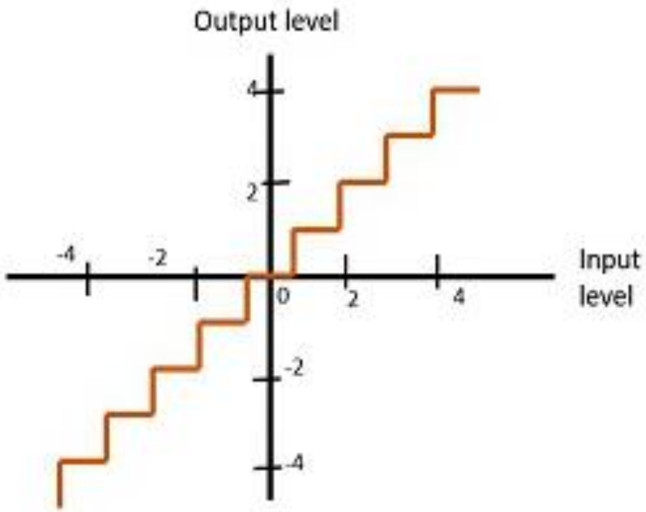


Fig 2 : Mid-Tread type Uniform Quantization

Multimedia Systems
Lecture – 18

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

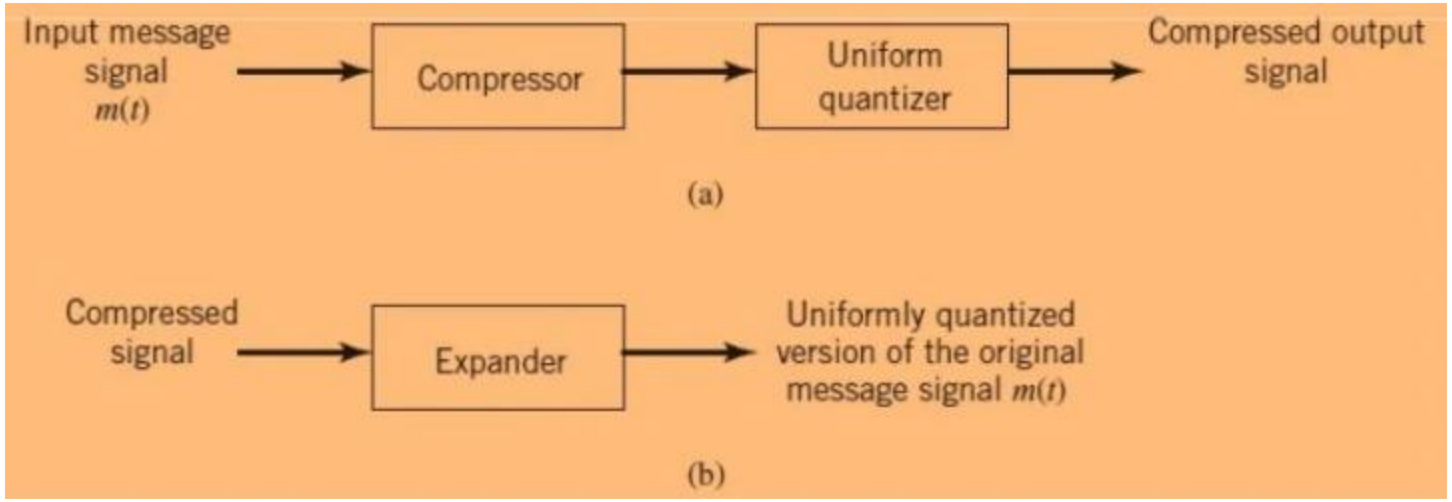
Non-uniform Quantization

- With a limited number of bits available, it may be more sensible to try to take into account the properties of human perception and set up nonuniform quantization levels that pay more attention to the frequency range over which humans hear best.
- If the quantization characteristic is nonlinear then the step size is not constant and quantization is known as non-uniform quantization.
- It is mostly used in case of speech or music as here the variation in amplitude is high which is expressed as crest factor and is given by

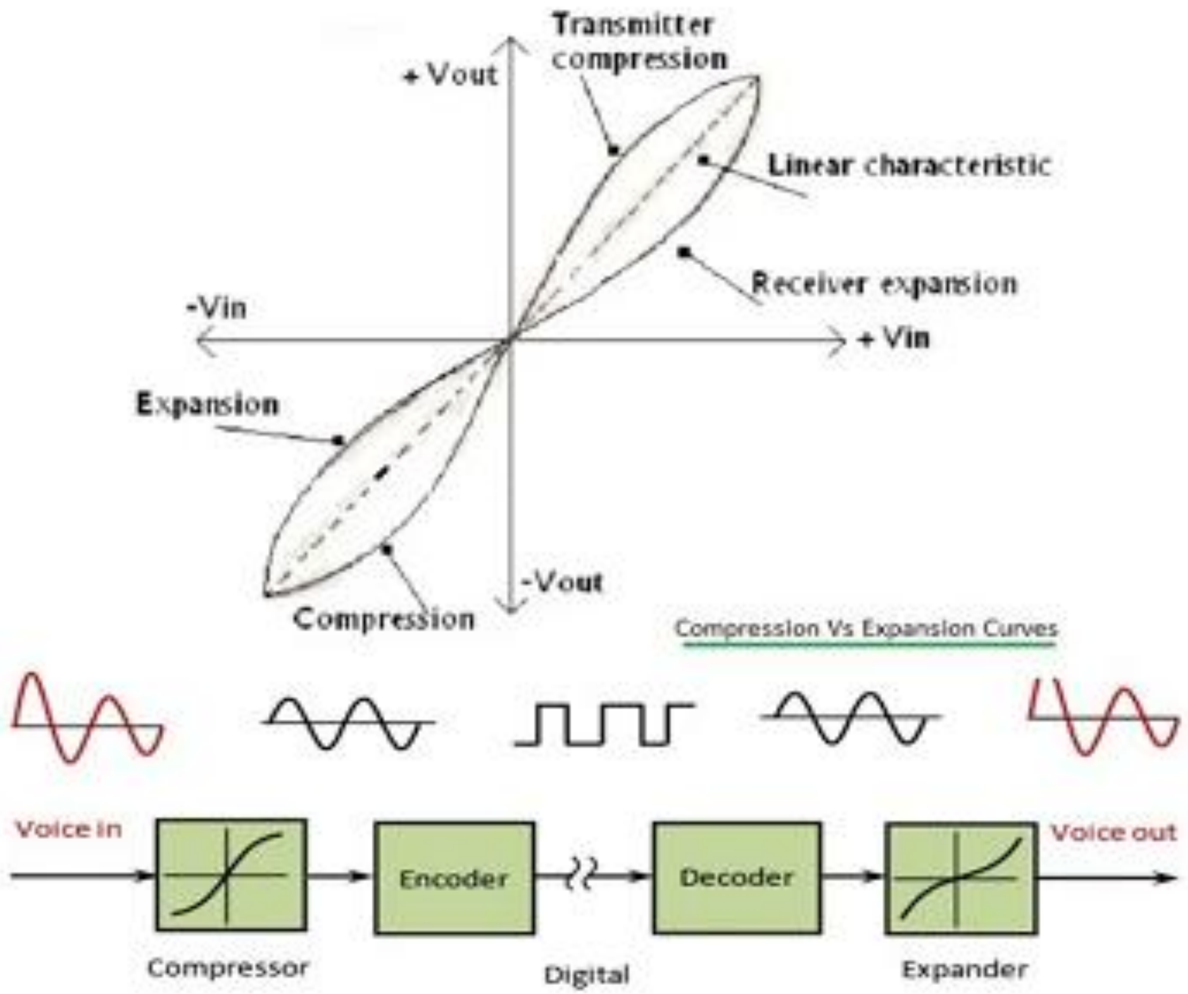
  crest factor = peak value of signal/rms value of signal

- Non-uniform quantization is achieved using companding.

- Companding:
- It is derived from two words, ***Compressing*** and ***Expanding***.
- The desired form of non-uniform quantization can be achieved by using compressor followed by a uniform quantizer.



Companding Process



μ-law and A-law companding

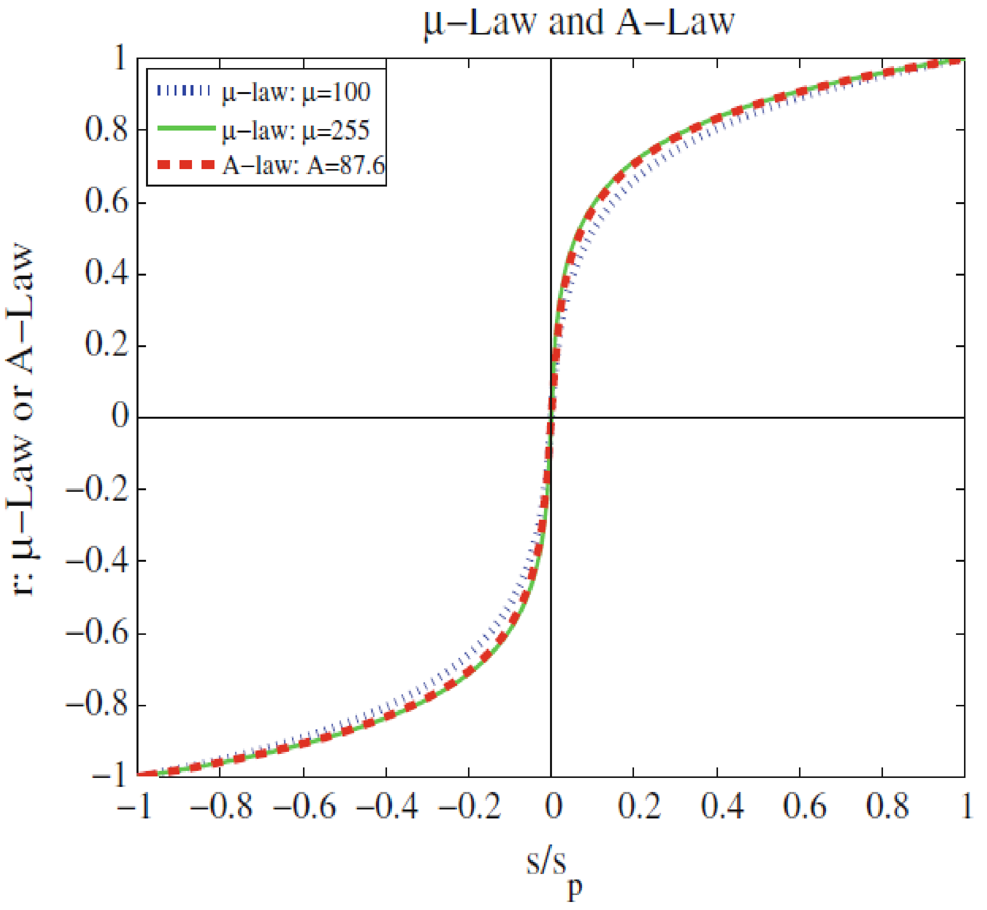
- μ-law is popular technique used in USA and Japan.
- Here the input and output relationship is given by

$$r = \frac{\operatorname{sign}(s)}{\ln(1+\mu)} \ln\left\{1 + \mu \left|\frac{s}{s_{\mathrm{p}}}\right|\right\}, \qquad \left|\frac{s}{s_{\mathrm{p}}}\right| \leq 1$$

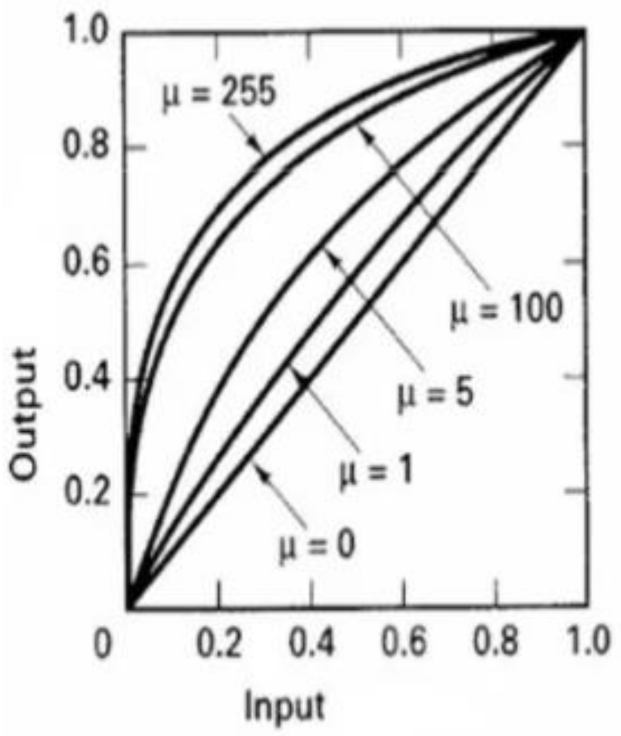
- A very similar rule, called *A-law*, is used in telephony in Europe.

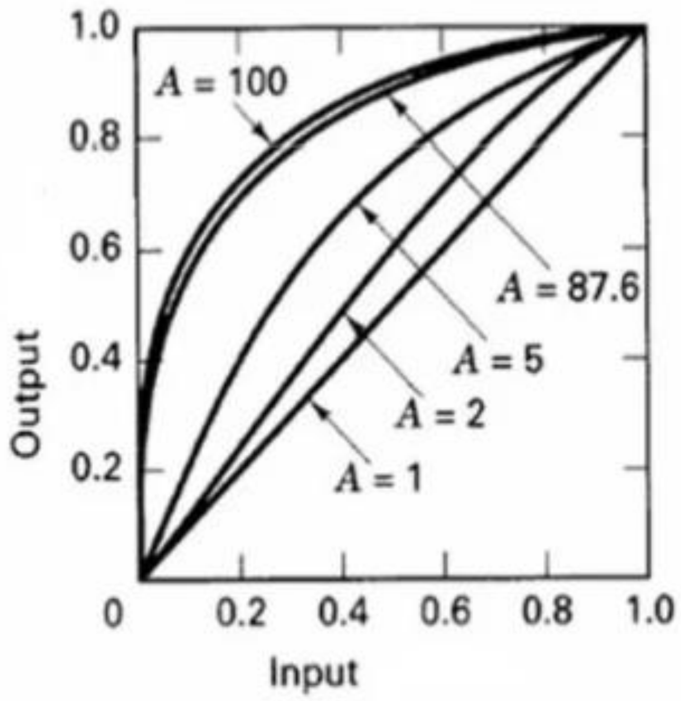
$$r = \begin{cases} \frac{A}{1+\ln A}\left(\frac{s}{s_{\mathrm{p}}}\right), & \left|\frac{s}{s_{\mathrm{p}}}\right| \leq \frac{1}{A} \\ \frac{\operatorname{sign}(s)}{1+\ln A}\left[1 + \ln A \left|\frac{s}{s_{\mathrm{p}}}\right|\right], & \frac{1}{A} \leq \left|\frac{s}{s_{\mathrm{p}}}\right| \leq 1 \end{cases}$$

$$\text{where } \operatorname{sign}(s) = \begin{cases} 1 & \text{if } s > 0, \\ -1 & \text{otherwise} \end{cases}$$


μ−Law and A−Law
μ−law: μ=100
μ−law: μ=255
A−law: A=87.6
r: μ−Law or A−Law
s/s
p
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1

μ-law and A-law Compression Characteristics

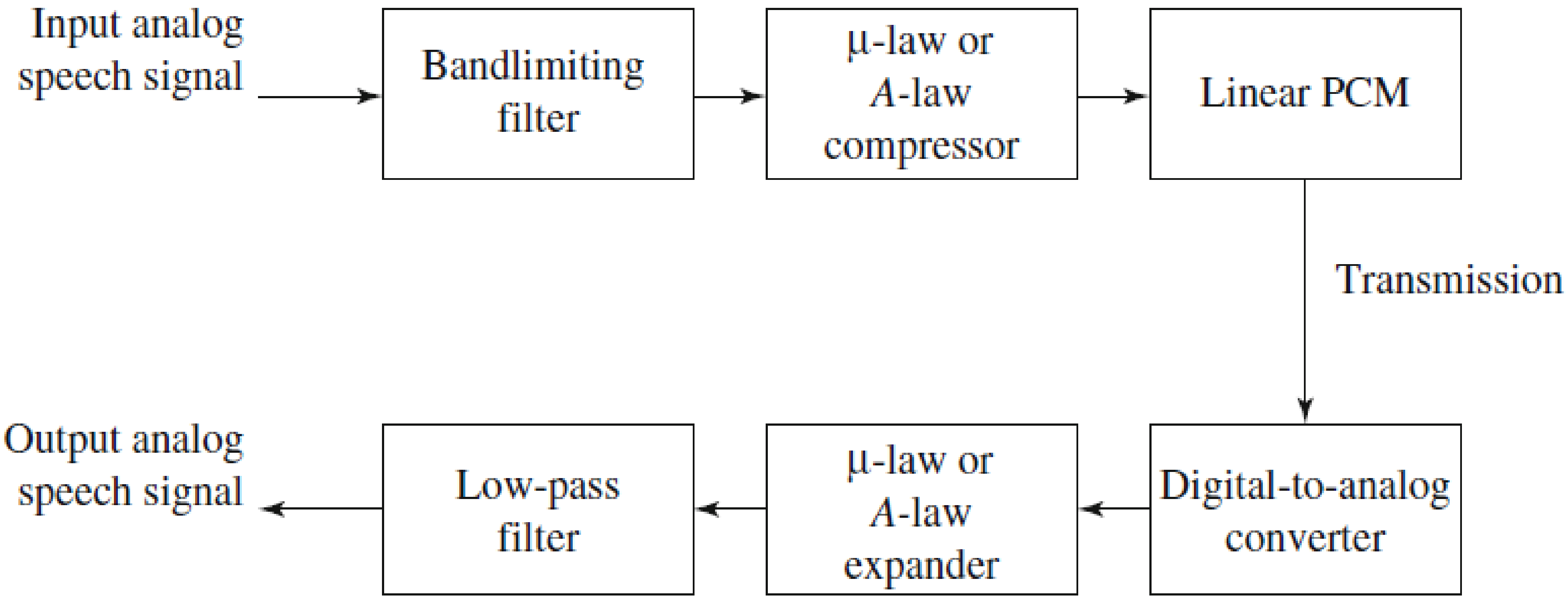


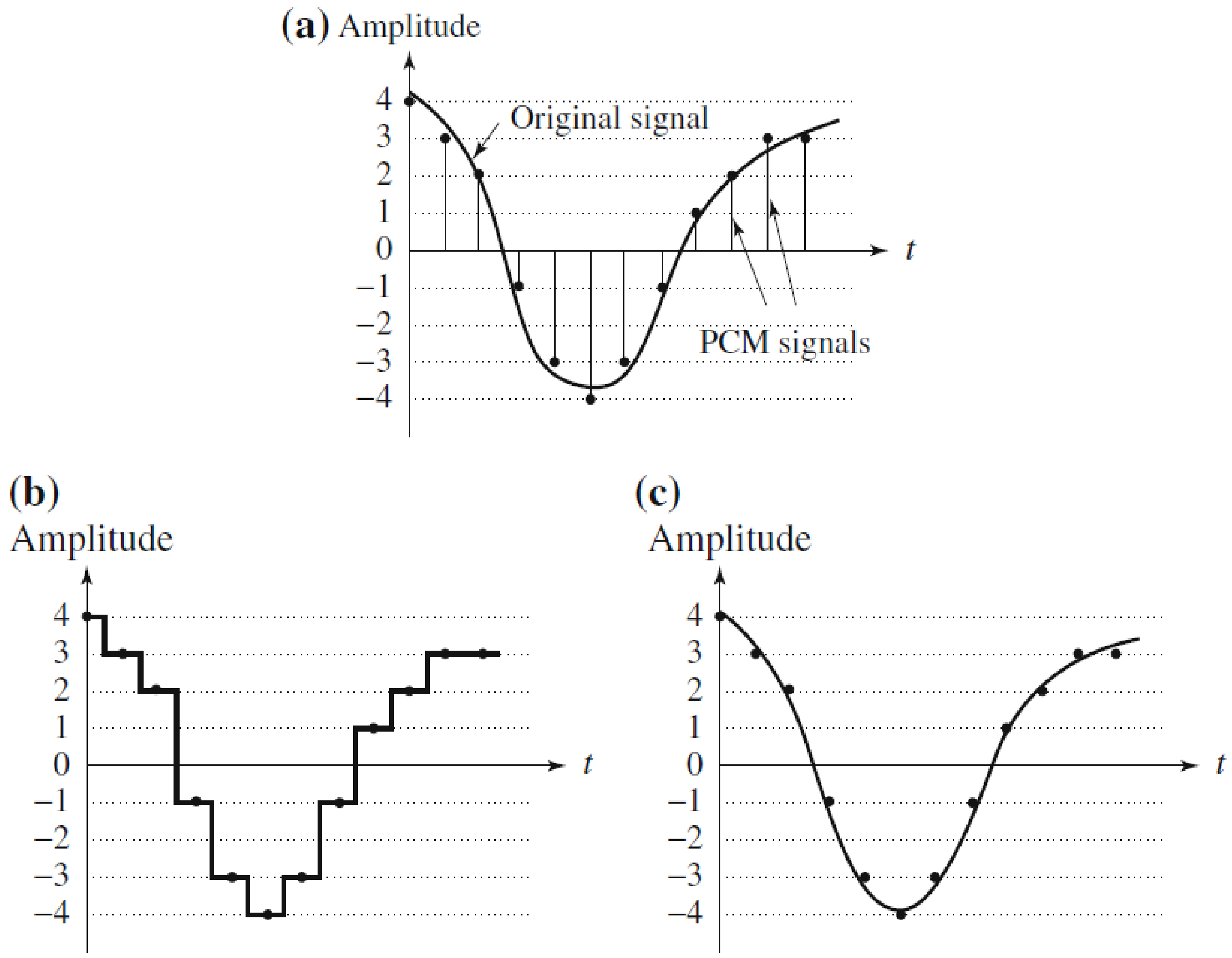


Pulse Code Modulation (PCM)

- **Modulation** is the process of varying one or more parameters of a carrier signal in accordance with the instantaneous values of the message signal.
- There are many modulation techniques, which are classified according to the type of modulation employed. Of them all, the digital modulation technique used is **Pulse Code Modulation.**
- We know that the basic techniques for creating digital signals from analog ones consist of *sampling* and *quantization*.
- Pulse Code Modulation, is a formal term for the sampling and quantization we have already been using.
- *Pulse* comes from an engineer's point of view that the resulting digital signals can be thought of as infinitely narrow vertical "pulses."

Basic Elements of PCM





a original analog signal and its corresponding PCM signals; **b** decoded staircase signal; **c** reconstructed signal after low-pass filtering

Multimedia Systems
Lecture – 19

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

Differential Pulse Code Modulation (DPCM)

- It is based on **differential predictive coding**.
- Audio is often stored not in simple PCM but in a form that ***exploits differences***.
- Generally, if a time-dependent signal has some consistency over time (*temporal redundancy*), the difference signal—subtracting the current sample from the previous one—will have a more ***peaked histogram***, with a maximum around zero.
- For a start, differences will generally be smaller numbers and hence offer the possibility of using fewer bits to store.

- Suppose our integer sample values are in the range 0 .. 255. Then differences could be as much as −255 .. 255. So we have unfortunately increased our *dynamic range* (ratio of maximum to minimum) by a factor of two.
- Let’s formalize our statement of what we are doing by defining the integer signal as the set of values f_n.
- Then we *predict* $\hat{f}_n$ values as simply the previous value, and we define the error e_n as the difference between the actual and predicted signals:

$$\hat{f}_n = f_{n-1}$$

$$e_n = f_n - \hat{f}_n$$

- We certainly would like our error value e_n to be as small as possible.
- Therefore, we would wish our prediction $\hat{f}_n$ to be as close as possible to the actual signal f_n.

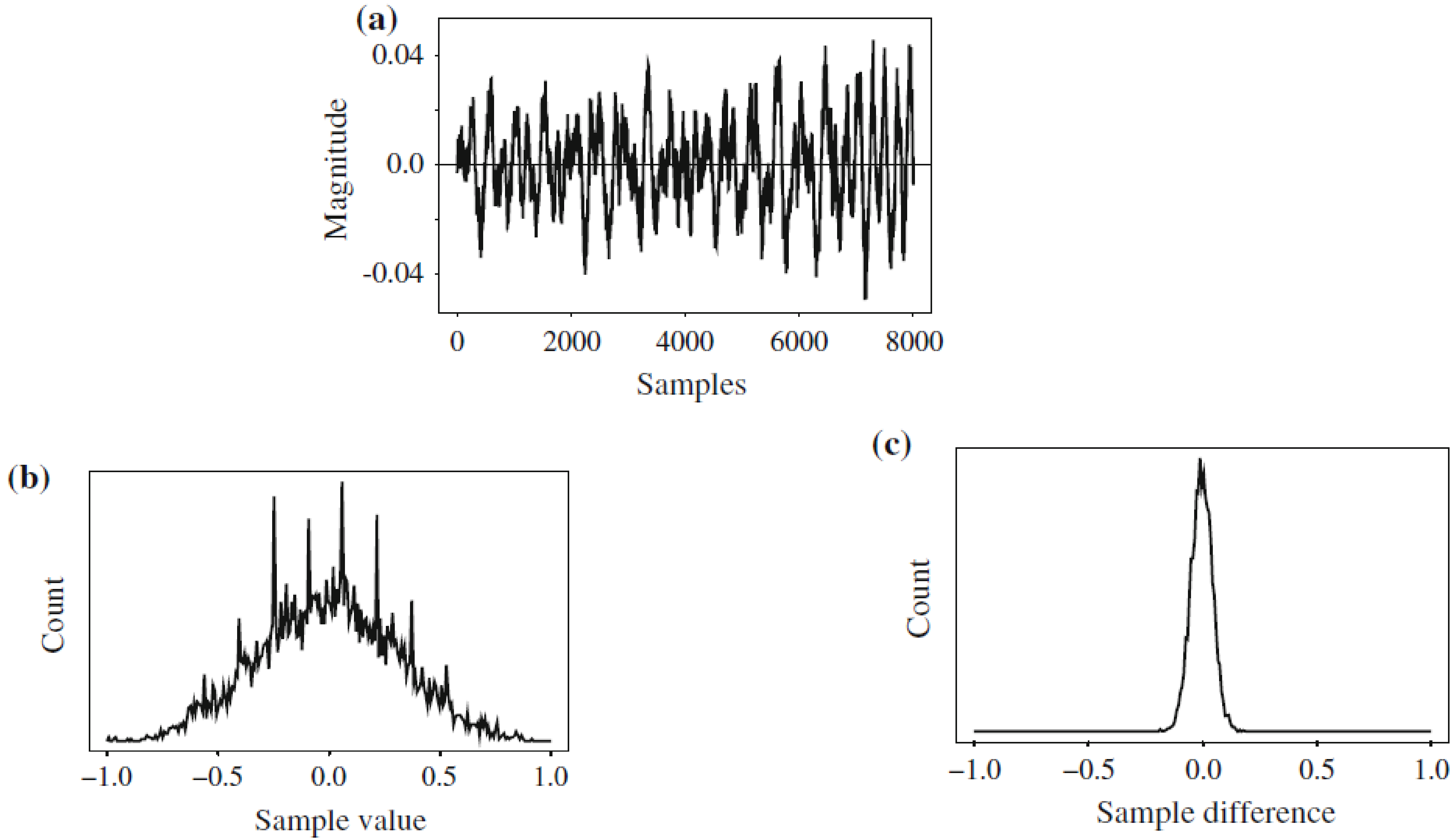
- But for a particular sequence of signal values, some *function* of a few of the previous values, f_{n-1}, f_{n-2}, f_{n-3}, etc., may provide a better prediction of f_n.

$$\hat{f}_n = \sum_{k=1}^{2 \text{ to } 4} a_{n-k} f_{n-k}$$

- The idea of forming differences is to make the histogram of sample values more peaked.

- ***So we can assign short codes to prevalent values like zeros here and long codewords to rarely occurring ones.***

Differencing concentrates the histogram: **a** digital speech signal; **b** histogram of digital speech signal values; **c** histogram of digital speech signal differences



- Example: suppose we devise a predictor for $\hat{f}_n$ as follows:

$$\hat{f}_n = \lfloor \frac{1}{2}(f_{n-1} + f_{n-2}) \rfloor$$
$$e_n = f_n - \hat{f}_n$$

- Then the error e_n (or a codeword for it) is what is actually transmitted. Suppose we wish to code the sequence

f1, f2, f3, f4, f5 = 21, 22, 27, 25, 22

- For the purposes of the predictor, we'll invent an extra signal value f_0, equal to $f1 = 21$, and first transmit this initial value, uncoded; after all, every coding scheme has the extra expense of some header information.

- Then the first error, $e1$, is zero, and subsequently

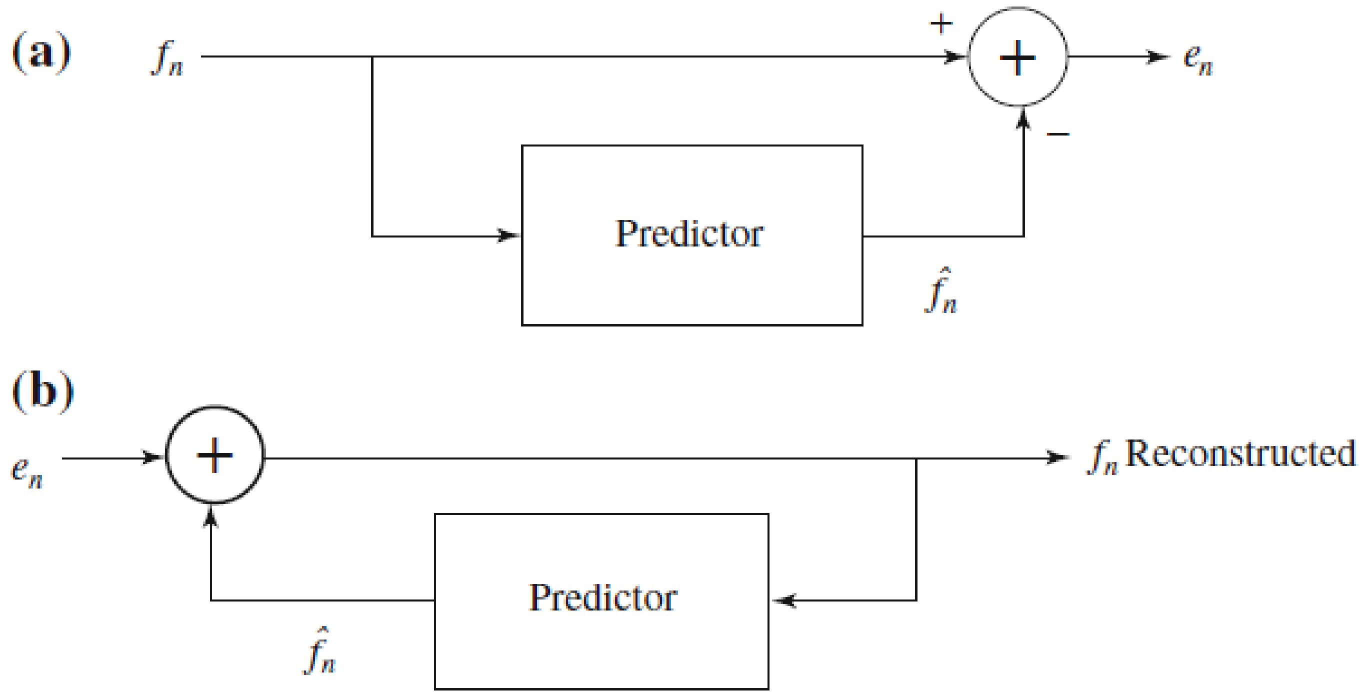
$$\hat{f}_2 = 21, \quad e_2 = 22 - 21 = 1$$

$$\hat{f}_3 = \lfloor \frac{1}{2}(f_2 + f_1) \rfloor = \lfloor \frac{1}{2}(22 + 21) \rfloor = 21$$
$$e_3 = 27 - 21 = 6$$

$$\hat{f}_4 = \lfloor \frac{1}{2}(f_3 + f_2) \rfloor = \lfloor \frac{1}{2}(27 + 22) \rfloor = 24$$
$$e_4 = 25 - 24 = 1$$

$$\hat{f}_5 = \lfloor \frac{1}{2}(f_4 + f_3) \rfloor = \lfloor \frac{1}{2}(25 + 27) \rfloor = 26$$
$$e_5 = 22 - 26 = -4$$

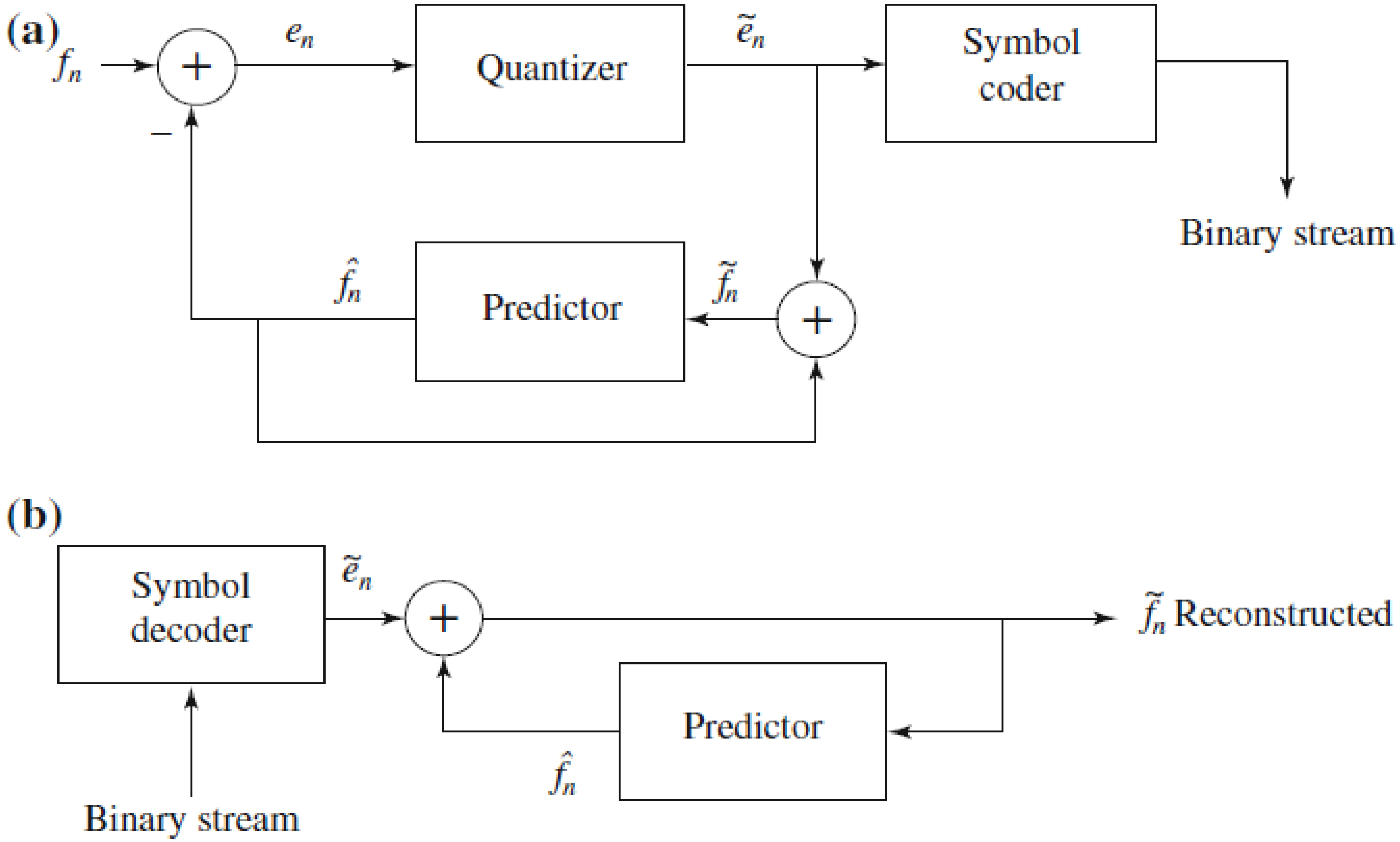
Schematic diagram for Predictive Coding: **a** encoder; **b** decoder



- ***Differential Pulse Code Modulation (DPCM)*** is exactly the same as Predictive Coding, Predictive coding except that it incorporates a ***quantizer step***.
- We shall call the original signal f_n, the predicted signal $\hat{f}_n$, and the quantized, reconstructed signal $\tilde{f}_n$. How DPCM operates is to form the prediction, from an error e_n by subtracting the prediction from the actual signal, then quantize the error to a quantized version, $\tilde{e}_n$.
- The equations that describe DPCM are as follows

$$
\begin{aligned}
\hat{f}_n &= \text{function\_of } (\tilde{f}_{n-1}, \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots) \\
e_n &= f_n - \hat{f}_n \\
\tilde{e}_n &= Q[e_n] \\
&\quad \text{transmit codeword}(\tilde{e}_n) \\
&\quad \text{reconstruct: } \tilde{f}_n = \hat{f}_n + \tilde{e}_n
\end{aligned}
$$

Schematic diagram for DPCM: **a** encoder; **b** decoder



- Codewords for quantized error values $\tilde{e}_n$ are produced using ***entropy coding***, such as Huffman coding.
- Notice that the predictor is always based on the reconstructed, quantized version of the signal $\tilde{f}_n$: the reason for this is that then the encoder side is not using any information not available to the decoder side.
- The main effect of the coder–decoder process is to produce $\tilde{f}_n$ reconstructed, quantized signal values $\tilde{f}_n = \hat{f}_n + \tilde{e}_n$
- The distortion is the average squared error $[\sum_{n=1}^{N}(\tilde{f}_n - f_n)^2]/N$
- The predictor makes use of the reconstructed, quantized signal values not actual signal values f_n—that is, the encoder simulates the decoder in the predictor path. The quantizer can be uniform or nonuniform.

- The prediction value $\hat{f}_n$ is based on however much history the prediction scheme requires: we need to buffer previous values of $\tilde{f}_n$ to form the prediction.
- Notice that the quantization noise, $f_n - \tilde{f}_n$ is equal to the quantization effect on the error term, $e_n - \tilde{e}_n$
- Example: Suppose we adopt a particular predictor as

$$\hat{f}_n = \text{trunc}\left[\left(\tilde{f}_{n-1} + \tilde{f}_{n-2}\right)/2\right]$$

so that $e_n = f_n - \hat{f}_n$ is an integer.

- the particular quantization scheme

$$\tilde{e}_n = Q[e_n] = 16 * \text{trunc}\left[(255 + e_n)/16\right] - 256 + 8$$
$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

- Suppose we wish to code the sequence $f1, f2, f3, f4, f5 = 130, 150, 140, 200, 230$.

- We prepend extra values $f_0 = 130$ in the datastream that replicate the first value, f_1, and initialize with quantized error $\tilde{e}_1 \equiv 0$, so that we ensure the first reconstructed value is exact: $\tilde{f}_1 = 130$.
- Then subsequent values calculated are as follows

$$
\begin{aligned}
\hat{f} &= \boxed{130},\ 130,\ 142,\ 144,\ 167 \\
e &= \boxed{0},\ 20,\ -2,\ 56,\ 63 \\
\tilde{e} &= \boxed{0},\ 24,\ -8,\ 56,\ 56 \\
\tilde{f} &= \boxed{130},\ 154,\ 134,\ 200,\ 223
\end{aligned}
$$

Delta Modulation (DM)

- It is a much-simplified version of DPCM often used as a quick analog-to-digital converter.

Uniform-Delta DM

- The idea in DM is to use only a *single* quantized error value, either positive or negative. Such a 1-bit coder thus produces coded output that follows the original signal in a staircase fashion.
- The relevant set of equations is as follows:

$$\hat{f}_{\mathrm{n}} = \tilde{f}_{\mathrm{n}-1}$$

$$e_{\mathrm{n}} = f_{\mathrm{n}} - \hat{f}_{\mathrm{n}} = f_{\mathrm{n}} - \tilde{f}_{\mathrm{n}-1}$$

$$\tilde{e}_{\mathrm{n}} = \begin{cases} +k \text{ if } e_{\mathrm{n}} > 0, & \text{where } k \text{ is a constant} \\ -k \text{ otherwise,} \end{cases}$$

$$\tilde{f}_{\mathrm{n}} = \hat{f}_{n} + \tilde{e}_{\mathrm{n}}$$

- Note that the prediction simply involves a delay.
- Example: Suppose signal values are as follows

f_1	f_2	f_3	f_4
10	11	13	15

- We also define an exact reconstructed value $\tilde{f}_1 = f_1 = 10$.
- Suppose we use a step value $k = 4$. Then we arrive at the following values:

$$
\begin{array}{llll}
\hat{f}_2 = 10, & e_2 = 11 - 10 = 1, & \tilde{e}_2 = 4, & \tilde{f}_2 = 10 + 4 = 14 \\
\hat{f}_3 = 14, & e_3 = 13 - 14 = -1, & \tilde{e}_3 = -4, & \tilde{f}_3 = 14 - 4 = 10 \\
\hat{f}_4 = 10, & e_4 = 15 - 10 = 5, & \tilde{e}_4 = 4, & \tilde{f}_4 = 10 + 4 = 14
\end{array}
$$

- We see that the reconstructed set of values 10, 14, 10, 14 never strays far from the correct set 10, 11, 13, 15.
- It is not difficult to discover that DM copes well with more or less constant signals, but not as well with rapidly changing signals.

Adaptive DM

- However, if the slope of the actual signal curve is high, the staircase approximation cannot keep up.
- A straightforward approach to dealing with a steep curve is to simply change the step size *k adaptively*—that is, in response to the signal's current properties.

Multimedia Systems
Lecture – 21

By

Dr. Priyambada Subudhi

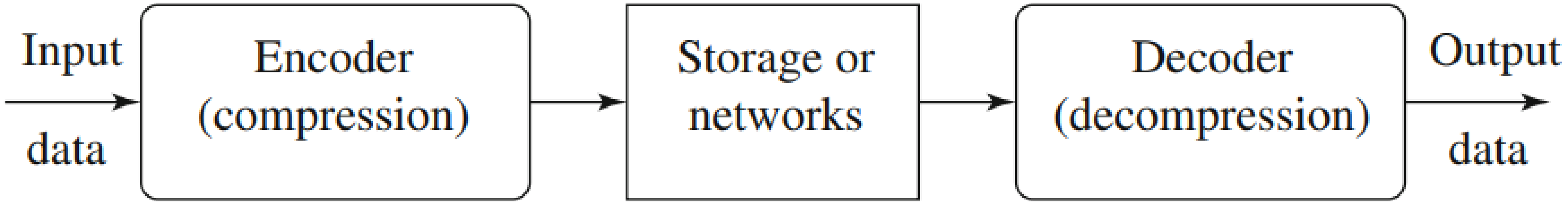
Assistant Professor

IIIT Sri City

Multimedia Data Compression

- The amount of digital media data that is produced in the form of text, video, audio, 3D graphics, and combinations of these media types is extraordinarily large, and the rate of creation increases every day.
- This growing mass of data needs to be stored, accessed, and delivered to a multitude of clients over digital networks, which have varying bandwidths.
- The existence of voluminous data, from creation to storage and delivery, motivates the need for compression.
- The role played in multimedia by data compression, perhaps the most important enabling technology that makes modern multimedia systems possible.

- In a general data compression scheme, in which compression is performed by an encoder and decompression is performed by a decoder.



- We call the output of the encoder *codes* or *codewords*.
- The intermediate medium could either be data storage or a communication /computer network.
- If the compression and decompression processes induce no information loss, the compression scheme is *lossless*; otherwise, it is *lossy*.

- Compression Ratio: If the total number of bits required to represent the data before compression is $\boldsymbol{B_0}$ and the total number of bits required to represent the data after compression is $\boldsymbol{B_1}$, then we define the compression ratio as

$$\text{compression ratio} = \frac{B_0}{B_1}.$$

- In general, we would desire any ***codec*** (encoder/decoder scheme) to have a compression ratio much larger than 1.0.

- The higher the compression ratio, the better the lossless compression scheme, as long as it is computationally feasible.

Basics of Information Theory

- When transmitting information information theory concerns itself with the efficiency and reliability of the transmission.
- It allows us to describe the process to compress data without losing its information content.
- The data could represent virtually anything— simple text, documents, images, graphics, and even binary executables.
- Here we will consider all these varied data types as generic data represented digitally in a binary form by a sequence of 1s and 0s.

- To understand compression, it is necessary to understand the information content of a message.
- Information relates to the organization in the data as a sequence of symbols. If the sequence changes, so does the information.
- ***Example:*** binary data stream of ***80,000*** bits.
- These bits might not need to be treated individually, but might instead be grouped into symbols.
- If it represent a gray-intensity image with each pixel represented by 8 bits, then we have ***10,000*** pixels or symbols.
- if width and height of the image are both 100, be interpreted as a gray image of size 100×100.
- Each symbol is represented by 8 bits, and there can be $2^8 = 256$ different possible gray levels or symbols.
- Also, the arrangement of the symbols is important

- Alphabet and Symbol: Information can be thought of as an organization of individual elements called *symbols*.
- An *alphabet* is defined as a distinct and nonempty set of symbols.
- The number or length of symbols in the set is known as the *vocabulary*.
- In the previous example involving a gray image, each pixel is represented by 8 bits and can have one of 2^8 or 256 unique values. Here, vocabulary consists of 256 symbols, where each symbol is represented or coded by 8 bits.
- Sequence: A series of symbols of a given alphabet form a sequence. For the alphabet {s1, s2, s3, s4}, a sample sequence is s1 s2 s1 s2 s2 s2 s1 s2 s3 s4 s1 s2 s3.
- A sequence of symbols is also termed a message produced by the source using the alphabet, and represents information.
- When looking at a sequence, some symbols might occur more commonly than other symbols. The frequency of occurrence of a symbol is an important factor when coding information represented by the symbols. ***The frequency is also known as probability***.

- Symbol Probability: The probability of occurrence of a symbol is defined by the ratio of the number of occurrences of that symbol over the length of the entire message.

$$P_i = \frac{m_i}{N},$$

- where m_i is the number of times symbol s_i occurs in the message of length N.
- Most coding algorithms make extensive use of symbol probabilities to obtain optimal codes for compression.

- **Entropy:** Quantifies the amount of information contained in a message of symbols given their probabilities of occurrence.
- For source-producing symbols, where each symbol s_i has a probability distribution P_i , the entropy is defined as

$$H = \sum P_i \log_2\left(\frac{1}{P_i}\right) = -\sum P_i \log_2 P_i.$$

- For the symbol s_i having a probability P_i , Shannon defined the notion of self-information of the symbol given by $\boldsymbol{\log_2(1/P_i)}$.
- The self-information represents number of bits of information contained in the symbol and, hence, the number of bits used to send that message.
- Entropy, then, becomes the ***weighted average of the information carried by each symbol*** and, hence, the ***average symbol length***.

- Example: If the probability of having the character n in a manuscript is 1/32, the amount of information associated with receiving this character is 5 bits.
- In other words, a character string nnn will require 15 bits to code.
- The definition of entropy is aimed at identifying often-occurring symbols in the datastream as good candidates for short codewords in the compressed bitstream.
- We use a variable-length coding scheme for entropy coding.

- if the information source S is a gray-level digital image, each si is a gray-level intensity ranging [0, 255], since 8 bits are typically used.
- Fig. a shows the histogram of an image with uniform distribution of gray-level intensities—that is, $\forall i\ p_i = 1/256$. Hence, the entropy of this image is

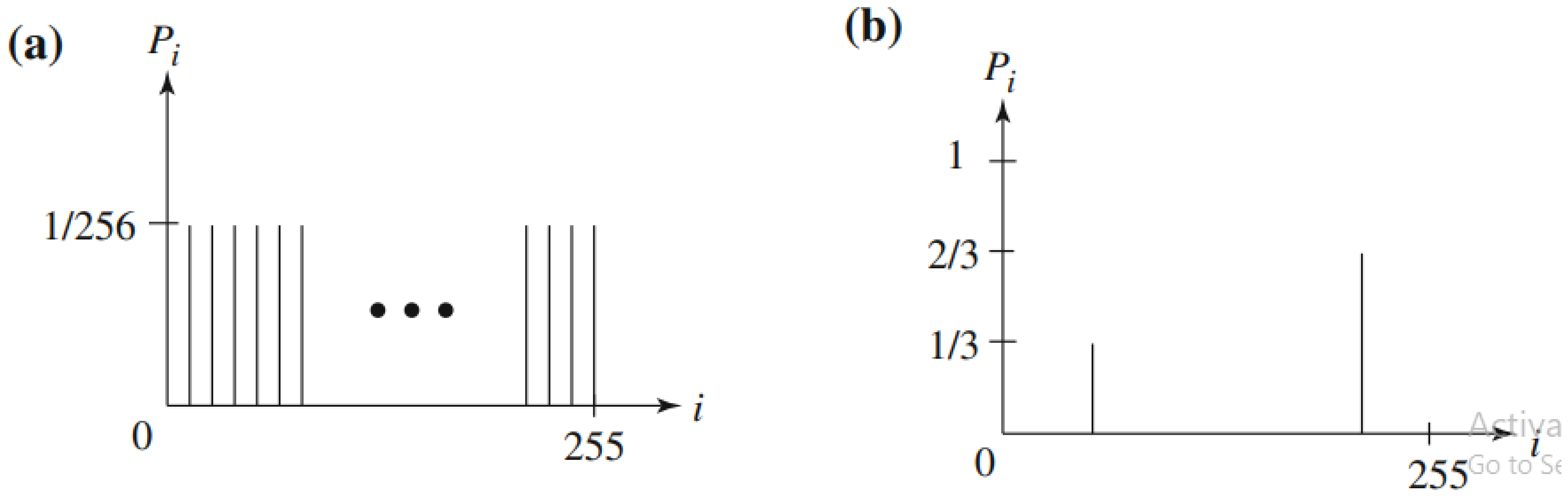
$$\eta = \sum_{i=0}^{255} \frac{1}{256} \cdot \log_2 256 = 256 \cdot \frac{1}{256} \cdot \log_2 256 = 8$$

- Fig. b shows the histogram of another image, in which 1/3 of the pixels are rather dark and 2/3 of them are rather bright. The entropy of this image is

$$\begin{aligned}\eta &= \frac{1}{3} \cdot \log_2 3 + \frac{2}{3} \cdot \log_2 \frac{3}{2} \\ &= 0.33 \times 1.59 + 0.67 \times 0.59 = 0.52 + 0.40 = 0.92\end{aligned}$$

- The entropy is greater when the probability distribution is flat and smaller when it is more peaked

Histograms for two gray-level images. a Uniform distribution; b A sample binary image



Multimedia Systems
Lecture – 21

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

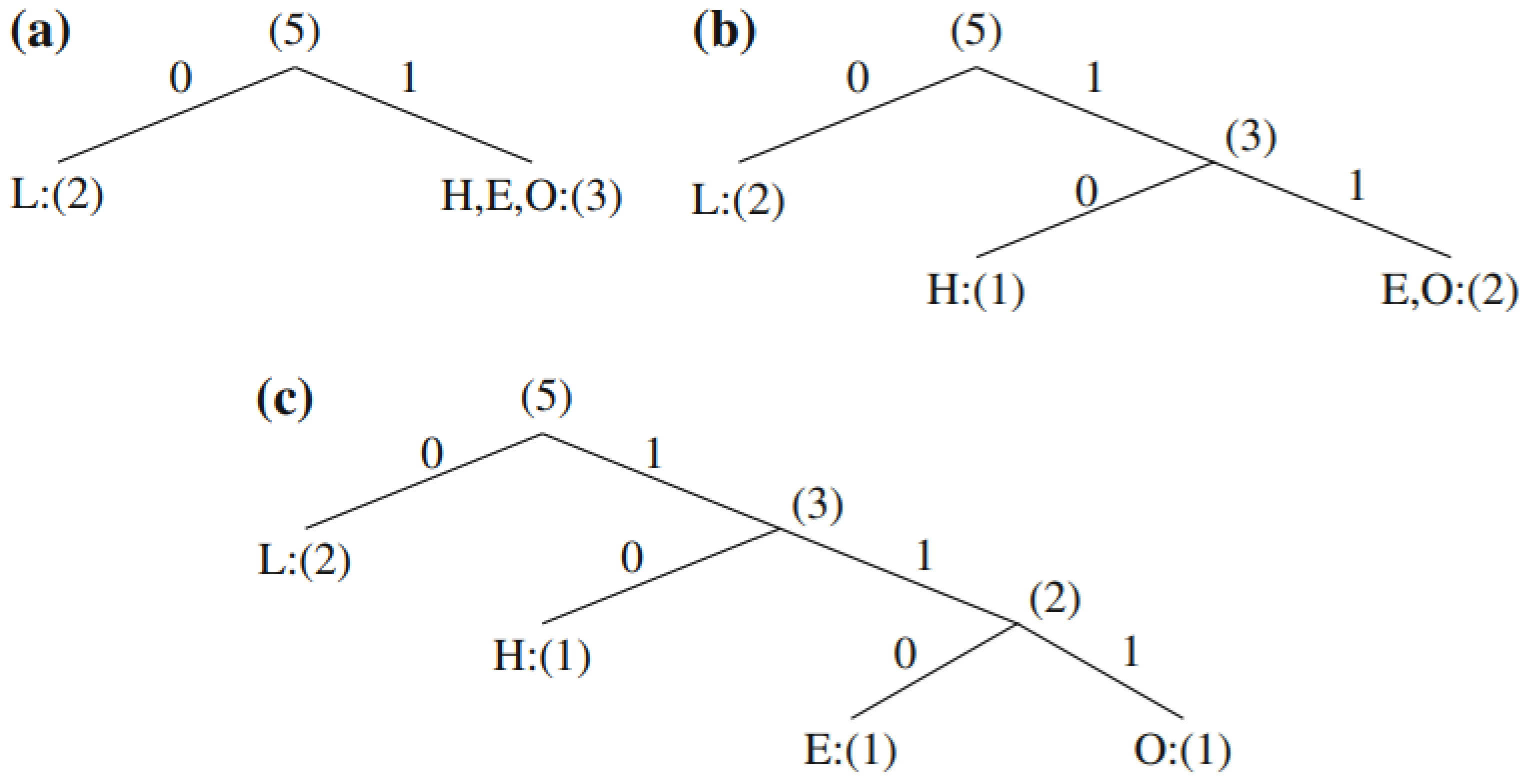
Variable-Length Coding

- Since the entropy indicates the information content in an information source S, it leads to a family of coding methods commonly known as entropy coding methods.
- Variable-length coding (VLC) is one of the best-known such methods.
- We will study the
  - Shannon–Fano algorithm
  - Huffman coding

Shannon-Fano Algorithm

- let us suppose the symbols to be coded are the characters in the word HELLO. The frequency count of the symbols is
- Symbol H E L O
- Count 1 1 2 1
- The encoding steps of the Shannon–Fano algorithm can be presented in the following top-down manner:
  - Sort the symbols according to the frequency count of their occurrences.
  - Recursively divide the symbols into two parts, each with approximately the same number of counts, until all parts contain only one symbol.

- Initially, the symbols are sorted as LHEO



One result of performing the Shannon–Fano algorithm on HELLO

Symbol	Count	$\log_2 \frac{1}{p_i}$	Code	Number of bits used
L	2	1.32	0	2
H	1	2.32	10	2
E	1	2.32	110	3
O	1	2.32	111	3
TOTAL number of bits:				10

- Entropy is given by

$$\eta = p_L \cdot \log_2 \frac{1}{p_L} + p_H \cdot \log_2 \frac{1}{p_H} + p_E \cdot \log_2 \frac{1}{p_E} + p_O \cdot \log_2 \frac{1}{p_O}$$
$$= 0.4 \times 1.32 + 0.2 \times 2.32 + 0.2 \times 2.32 + 0.2 \times 2.32 = 1.92$$

Huffman Coding

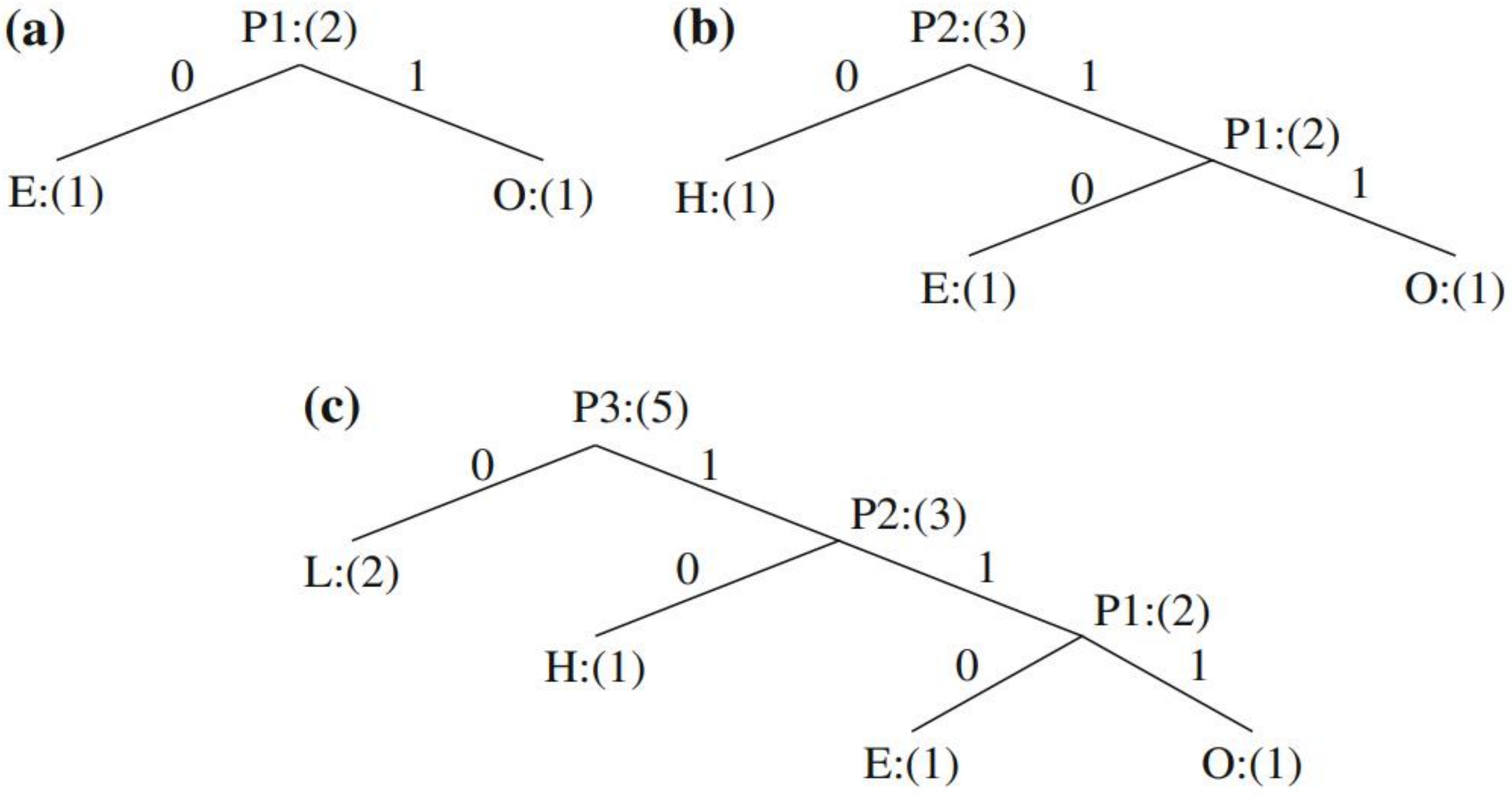
- First presented by Huffman in a 1952.
- This method attracted an overwhelming amount of research and has been adopted in many important and/or commercial applications, such as fax machines, JPEG, and MPEG.
- In contradistinction to Shannon–Fano, which is top-down, the encoding steps of the Huffman algorithm is a bottom-up approach.
- Let us use the same example word, HELLO. A similar binary coding tree will be used as above, in which the left branches are coded 0 and right branches 1. A simple list data structure is also used.

Huffman Coding Algorithm

Algorithm 7.1 (Huffman Coding).

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

Coding tree for HELLO using the Huffman algorithm. a First iteration; b Second iteration; c Third iteration



- symbols P1, P2, P3 are created to refer to the parent nodes in the Huffman coding tree.
- The contents in the list are illustrated below:
  - After initialization: L H E O
  - After iteration (a): L P1 H
  - After iteration (b): L P2
  - After iteration (c): P3
- The average number of bits used to code each character is also 2, (i.e., $(1 + 1 + 2 + 3 + 3)/5 = 2$).
- Example-2: Consider a text string containing a set of characters and their frequency counts as follows: A:(15), B:(7), C:(6), D:(6) and E:(5). Find out the number of bits required in case of Shanon-Fano and Huffman Coding.

Run Length Encoding

- Instead of assuming a memoryless source, run-length coding (RLC) exploits memory present in the information source.
- It is one of the simplest forms of data compression.
- The basic idea is that if the information source we wish to compress has the property that symbols tend to form continuous groups, instead of coding each symbol in the group individually, we can code one such symbol and the length of the group.
- A sample string of symbols is as follows:

  BBBBEEEEEEEECCCCDAAAAA.

- It can be represented as

  4B8E4C1D5A.

- Run length encoding is used in a variety of tools involving text, audio, images, and video.
- Consider a bilevel image (one with only 1-bit black and white pixels) with monotone regions. This information source can be efficiently coded using run-length coding. In fact, since there are only two symbols, we do not even need to code any symbol at the start of each run. Instead, we can assume that the starting run is always of a particular color (either black or white) and simply code the length of each run.