Module No 2

Text and Digital image

Text and Digital image

- Introduction

In computer science and information, data compression is the process of encoding information using fewer bits(or information bearing units) than an unencoded representation would use, through the use of specific encoding schemes.

In computing data reduplication is a specialised data compression technique for eliminating redundant data, typically to improve storage utilisation.

Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth.

On the downside compresses data must be decompressed to be used and this extra processing may be detrimental to some application.

The design of data compression schemes therefore involves trade offs among various factors, including the degree of compression, the amount of distortion introduced and the computational resources required to compress and decompress the data.

Introduction to Digitization Principle

Digitization is the representation of an object, image, sound, document or a signal by a discrete set of its points or samples. The result is called digital representation for the object and digital form for the signal.

The term digitization is often used when diverse forms of information such as text, sound, image or voice are converted into a single binary code. Digital information exists as one of two digits, either 0 or 1. These are know as bits and the sequence of 0’s and 1’s that constitute information are called bytes.

Digitization occurs in 2 parts :

Digitization : The reading of an analog signal A and, at regular time intervals(frequency), sampling the value of the signal at that point. Each such reading is called a sample and may be considered to have infinite precision at this stage

Quantization : Samples are rounded to a fixed set of numbers (such as integers), a process known as quantization. A series of digital integers can be transferred into an analog output that approximates the original analog signal. Such a transformation is called a DA conversion. The sampling rate and the number of bits used to represent the integers combine to determine how close such approximation to the analog signal a digitization will be.

Text

- Document digitization is the process of converting manual documents into digital formats. In the process of document digitizing any type of document like texts, images, video, business cards or periodicals are digitized and converted into digital formats such as text, html, xml, pdf, doc, xls, jpeg or tiff.
- Text File Formats

A file format is a particular way that information is encoded for storage in a computer file.

Different kinds of file formats for various kinds of information

e.g. Word processor documents have various file formats.

Multimedia File formats :

1. Rich text format (RTF)
2. Tagged image file format (TIFF)
3. Resource interchange file format (RIFF)

--- RIFF DIB file format

--- RIFF PAL file format

Rich Text Format (RTF)

From perceptive of multimedia, RTF is important because it is used to attach, embed or link other text files or link other text files or even binary files such as executable, audio/video files.

Text only field can be used for indexing purpose

The Rich Text File (RTF) specification is a method of encoding formatted text and graphics for easy transfer between applications.

The RTF specification provides a format for text and graphics interchange that can be used with different output devices, operating environments and OS.

RTF uses ANSI, PC-8, Macintosh or IBM PC character set to control the representation and formatting of a document, both on the screen and in print.

With RTF Specification documents created under different operating systems and with different software applications can be transferred between those OS and applications.

This file format allows embedding graphics and other file formats within a document. This format is used by products such as Lotus Notes. This format is also the basis for the use of OLE.

RTF

- Text editor carry information in American Standard Code for Information Interchange (ASCII) or Extended Binary Coded Decimal Interchange Code (EBCDIC) form, but they don't carry formatting information. RTF is a solution for which it carries formatting info. Also.
- The key information carried across in RTF format document file include :
- 1. Character Set : Common examples of character encoding systems include the ASCII and EBCDIC. ASCII is a 7-bit encoding scheme used to encode letters, numerals, symbols and device control codes as fixed length codes using integers. EBCDIC is an 8-bit encoding scheme developed by IBM.
- 2. Font Table : It contains the information for registering font files with the system
- 3. Colour Table : It lists colours used in the text for highlighting text. The colour table also mapped for display.
- 4. Document formatting : RTF provides true document margins. Paragraphs are specified relative to document margin. Printed paper looks like original page.
- 5. Section formatting : Section breaks store the setting for the section. User can select, delete, copy, or move these settings as well.
- 6. General formatting : Footnotes, annotations, bookmarks and pictures.
- 7. Character formatting : Formatting information such as B, *I*, <u>U</u> , strikethrough, shadow text, online text and hidden text are specified using control characters.
- 8. Special Characters : These includes hyper, spaces, backslashes etc.

Tagged Image File Format (TIFF)

- TIFF File Format TIFF is an industry-standard file format designed to represent raster image data generated by scanners, frame grabbers, and paint/ photo retouching applications.
- TIFF Version 6.0 .
- It offers the following formats: (i) Grayscale, palette colour, RGB full-colour images and black and white.
- (ii) Run-length encoding, uncompressed images and modified Huffman data compression schemes.
- The additional formats are: (i) Tiled images, compression schemes, images using CMYK, YCbCr colour models.
- TIFF Structure TIFF files consists of a header.
- The header consists of byte ordering flag, TIFF file format version number, and a pointer to a table.
- The pointer points image file directory. This directory contains table of entries of various tags and their information.

TIFF Structure

A TIFF file consists of 8 byte header containing flowing fields :

--- Byte ordering flag (Intel or Motorola) (2 byte)

--- TIFF file format version number (2 byte)

--- Pointer to the table called an “Image File Directory” (IFD) (4 byte)

1. Byte ordering flag (2 Byte)

Intel byte order

D	I Byte
C	II Byte
B	III Byte
A	IV Byte

Motorola byte order for 32 bit word ABCD

A	I Byte
B	II Byte
C	III Byte
D	IV Byte

This flag stores byte order. Byte order can be Intel byte order or Motorola byte order.

For Intel byte order first 2 bytes order 0 x 49 which represents ASCII character “II”.

For Motorola byte order, first 2 bytes order 0 x 4D which represents ASCII character “MM”.

TIFF Structure

2. Version number : This represents version number of TIFF file format specification. It may be version 5 or 6

3. Pointer to Image file directory (IFD) : The last 4 byte contain Pointer to IFD.

Fig. shows header structure for TIFF File.

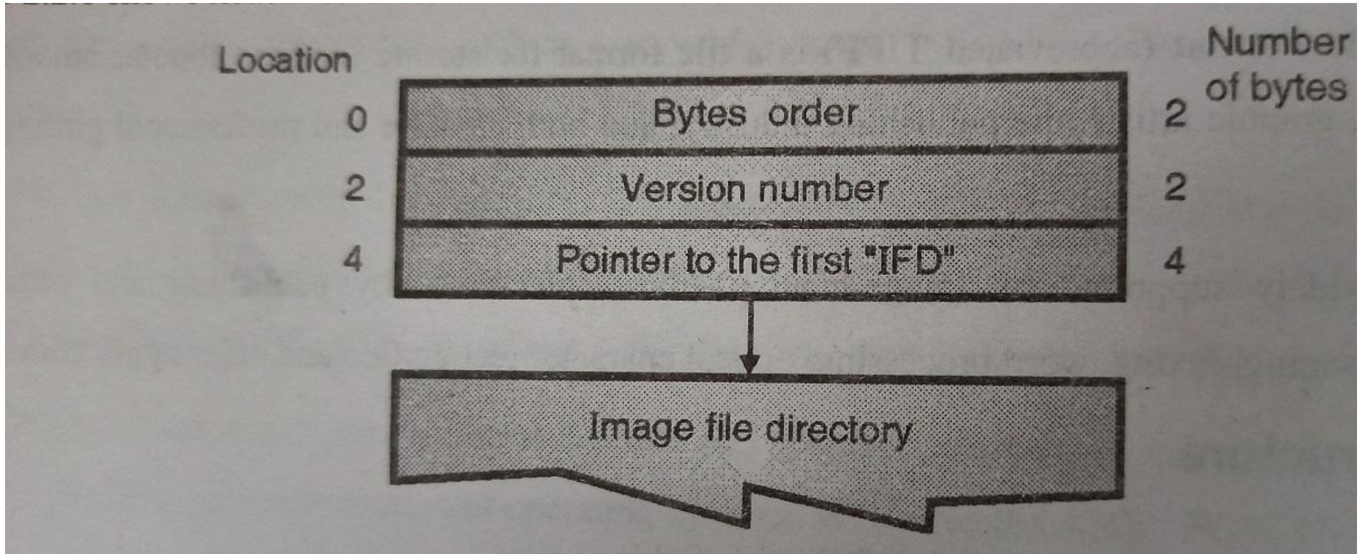


Fig. 2.2.1 : TIFF file format header

Structure of TIFF Image File Directory

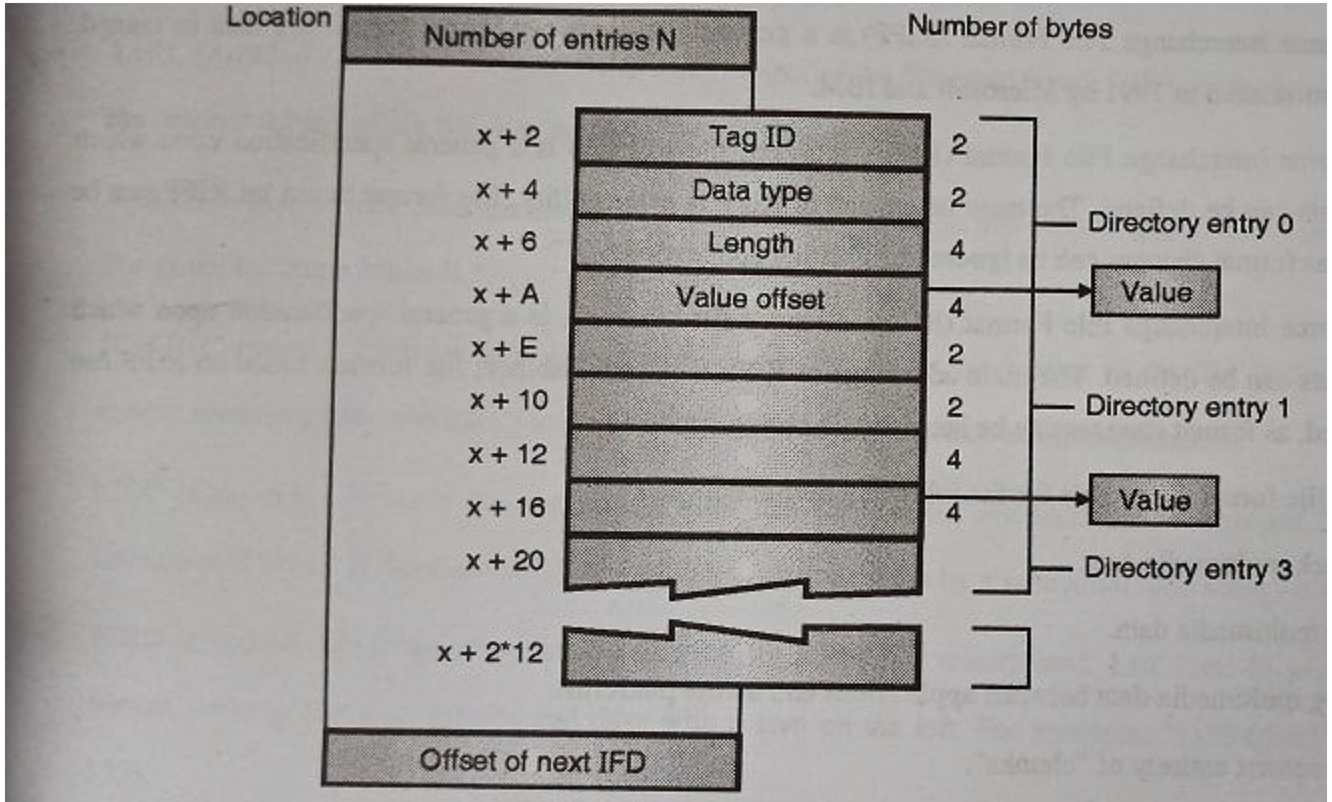


Fig. 2.2.2 : Structure of TIFF image file directory

IFD

IFD contains a table of entries of the various tags in the file and their associated information. IFD and its contents. The IFD is variable length table containing directory entries. Length of table depends on number of directory entries in the table.

The first two bytes contain total number of entries in table followed by directory entry.

Each directory entry consists of 12 bytes that is next 12 bytes represent the first directory entry and the next 12 byte after that represent second directory entry. The last item in IFD is four byte pointer that points to next IFD.

The byte content of each directory entry are as follows :

- The first two bytes of directory entry contain tag number-tag ID.
- Second two bytes represent the type of data. The byte values for supported data type.
- Next four bytes contain the length of (or count) for data type.
- Final four byte contains data or Pointer..

Contd ….

TIFF Tags

- TIFF Tags The first two bytes of each directory entry contain a field called the Tag ID.
- Tag IDs arc grouped into several categories.
- They are Basic, Informational, Facsimile, Document storage and Retrieval.
- TIFF Classes: (Version 5.0)
- It has five classes
- 1. Class B for binary images
- 2. Class F for Fax
- 3. Class G for grey-scale images
- 4. Class P for palette colour images
- 5. Class R for RGB full-colour images.

Resource Interchange File Format (RIFF)

The RIFF is a generic file container format for storing data in tagged chunks. It was introduced in 1991 by Microsoft and IBM.

In RIFF, a tagged file structure is a general specification upon which many file formats can be defined. The main advantage of RIFF is extensibility ; file format based on RIFF can be future proofed as format changes can be ignored by existing application.

The RIFF file format is suitable for the following multimedia tasks :

- Playing back multimedia data.
- Recording multimedia data.
- Exchanging multimedia data between applications and across platforms.
- RIFF file consist entirely of “chunks”

All Chunks have the following format

- 4 bytes : An ASCII identifier for this chunk e.g. “fmt” or “data”.
- 4 bytes : An unsigned, little endian 32 bit integer with the length of this chunk (except this field itself and the chunk identifier)
- Variable-sized field : The chunk data itself of the size given in the previous field.
- A pad byte if the chunk’s length is not given.

Resource Interchange File Format (RIFF)

The RIFF file formats consist' of blocks of data called chunks.

They are RIFF Chunk - defines the content of the RIFF file.

List Chunk - allows to embed archival location copy right information and creating date.

Sub chunk - allow additional information to a primary chunk

The first chunk in a RIFF file must be a RIFF chunk and it may contain one or more sub chunk

The first four bytes of the RIFF chunk data field are allocated for the form type field containing four characters to identify the format of the data stored in the file: AVI, WAV, RMI, PAL and so

The sub chunk contains a four-character ASCII string 10 to identify the type of data.

Four bytes of size contains the count of data values, and the data.

The data structure of a chunk is same as all other chunks.

Resource Interchange File Format (RIFF)

The <list-type> is a four-character code that identifies the contents of the list. If an application recognizes the list type, it should know how to interpret the sequence of subchunks. Microsoft has only documented one list type called INFO. Each INFO list chunk consist of four character ASCII string ID called tag and four byte field containing size of chunk data, followed by data for that chunk.

An 'INFO' list should contain only the following chunks.

- IARL (*Archival Location*): Indicates where the subject of the file is archived. IART *Artist.* Lists the artist of the original subject of the file. For example, "Michaelangelo."
- ICMS (*Commissioned*). Lists the name of the person or organization that commissioned the subject of the file. For example, "Pope Julian II."
- ICMT (*Comments*) Provides general comments about the file or the subject of the file. If the comment is several sentences long, end each sentence with a period. Do *not* include newline characters.
- ICOP (*Copyright*). Records the copyright information for the file. For example, "Copyright Encyclopedia International 1991." If there are multiple copyrights, separate them by a semicolon followed by a space.
- ICRD (*Creation date*). Specifies the date the subject of the file was created. List dates in year-month-day format, padding one-digit months and days with a zero on the left. For example, "1553-05-03" for May 3, 1553.
- **Sub chunk :** Allows adding more information to primary chunk when primary chunk is not sufficient. It contains a four-character ASCII string i.e. tag ID to identify the type of data , four bytes of size containing the count of data value and data itself.

RIFF

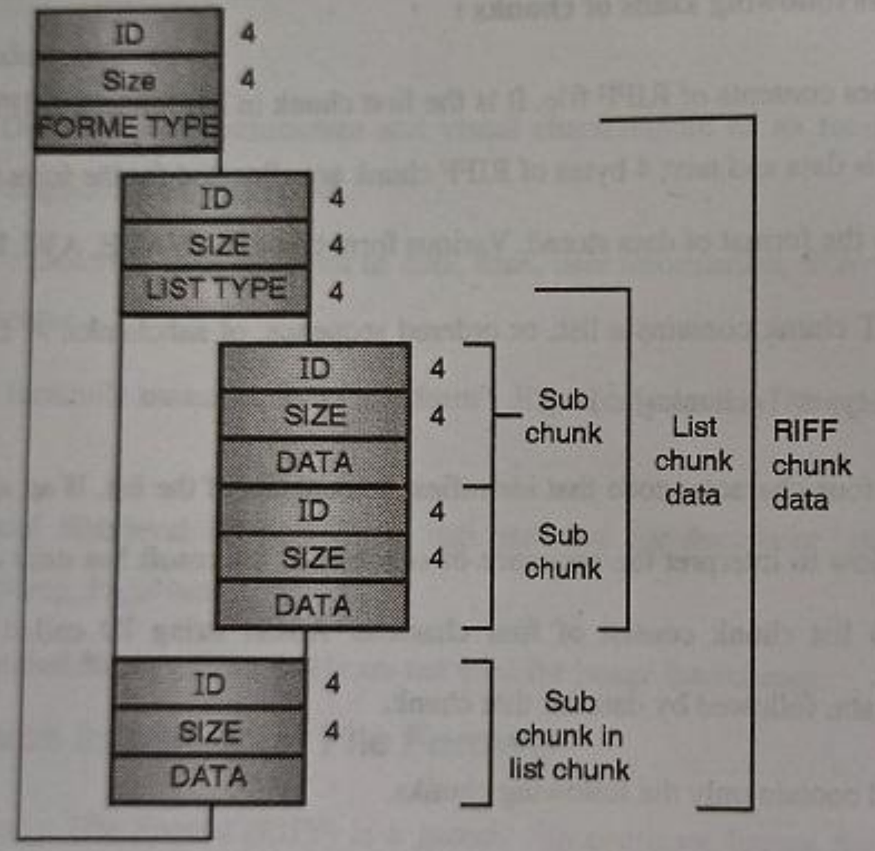


Fig. 2.2.3 : RIFF chunk with list chunk

RIFF-DIB

2.2.3(A) RIFF Device-Independent Bitmap File Format

The Device Independent Bitmap (DIB) format represents bitmap images in a device-independent manner. Bitmaps can be represented at 1, 4, and 8 bits per pixel, with a palette containing colors represented in 24 bits. Bitmaps can also be represented at 24 bits per pixel without a palette and in a run-length encoded format.

We will describes two types of RIFF Device-Independent Bitmap (RDIB) format :

- A simple RDIB consisting of a DIB file enclosed in a RIFF chunk.
- An extended RDIB that allows the creation of more complex bitmaps

To ensure that the maximum number of programs will accept an RDIB file, programs that adopt the extended RDIB format should also accept simple RDIB files. Both formats are described in the following sections.

1. Simple RDIB Format

The simple RDIB format consists of a Windows 3.0 or Presentation Manager 1.2 DIB enclosed in a 'RIFF' chunk. Enclosing the DIB in a 'RIFF' chunk allows the file to be consistently identified; for example, an 'INFO' list can be included in the file.

The simple 'RDIB' form is defined as follows, using the standard RIFF form definition notation : DIB files; RDIB format RIFF files; RDIB form RDIB file format File formats; RIFF Device-Independent Bitmap (RDIB)RDIB files; description

<RDIB-form> Ý RIFF ('RDIB' data(<DIB-data>))

Extended RDIB format

The **<DIB-data>** format is defined in "Device Independent Bitmap File Format," earlier in this chapter.

2. Extended RDIB Format

The extended RDIB format, designed to incorporate enhancements such as compression, is defined as follows:

```
<RDIB-form> Ý
RIFF('RDIB'
<bmhd-ck> // Bitmap header chunk
[ <pal-file> | // Internal palette chunk
<XPAL-ck> ] // External palette chunk
<bitmap-data> ) // Bitmap data
```

The **<pal-file>** chunk can be any of the palette-file formats discussed in "Palette File Format," later in this chapter. The **<bmhd-ck>**, **<XPAL-chunk>**, and **<bitmap-data>** are described in the following sections.

RIFF Palette File Format

2.2.3(B) RIFF Palette File Format

The Palette (PAL) File Format represents a logical palette, which is a collection of colors represented as RGB values.

The simple PAL format is defined as follows : Palette files; PAL file format PAL file format File formats; Palette (PAL)

RIFF('PAL' data(<palette:LOGPALETTE>))

LOGPALETTE is the Windows 3.0 logical palette structure, defined as follows : Palette files; logical palette structure LOGPALETTE data structure; with palette files Logical palette structure

```
typedef struct tagLOGPALETTE {
WORD palVersion;
WORD palNumEntries;
PALETTEENTRY palPalEntry[];
} LOGPALETTE;
```

The LOGPALETTE structure fields are as follows :

Field	Description
palVersion	Specifies the Windows version number for the structure.
palNumEntries	Specifies the number of palette color entries.
palPalEntry[]	Specifies an array of PALETTEENTRY data structures that define the color and usage of each entry in the logical palette.

RIFF Palette File Format

The colors in the palette entry table should appear in order of importance. This is because entries earlier in th logical palette are most likely to be placed in the system palette.

The PALETTEENTRY data structure specifies the color and usage of an entry in a logical color palette. Th structure is defined as follows : PALETTEENTRY data structure; with palette files Palette files; setting intensi RGB values; palette intensity

```
typedef struct tagPALETTEENTRY {
BYTE peRed;
BYTE peGreen;
BYTE peBlue;
BYTE peFlags;
} PALETTEENTRY;
```

TIFF Vs RIFF

2.2.4 TIFF Vs RIFF

Sr. No.	TIFF	RIFF
1.	TIFF format is developed by Aldus Corporation.	RIFF format is jointly developed by IBM and Microsoft.
2.	It is totally new format.	It is not new format it just provides wrapper around existing file formats to provide platform independence.
3.	It stores only bitmap data.	It can store other data also.
4.	TIFF includes number of compression scheme that allows developers to choose the best space or time trade-off for their application.	It includes Microsoft proprietary compression algorithms.
5.	Suitable for pre-press application.	Not suitable for pre-press application.
6.	Here tags are used for random access.	Here list chunks are used for random access. For example "idxl" list chunk of AVI RIFF file format is used for random access.
7.	Information is stored in Image file directory.	Information is stored in chunks.
8.	It contains 8 byte header to identify TIFF file.	It contain RIFF chunk to identify RIFF file.

Digital Image

Syllabus Topic : Digital Image

2.3 Digital Image

Digital images are commonly obtained by the following methods. The first method is to digitize photographs and printed pictures using a scanner. The digitization principle is similar to that of digitizing audio as described in the previous section. In digital cameras, ADCs are built in. The second source of digital images is from individual frames of digitized video. In the third method, images are generated by some sort of graphics (painting) package. In raw (bitmap) format, all digital images are represented in the same way regardless of the different original sources. In this section we describe how digital images are represented, the main parameters used to describe digital images, and common image compression techniques and standards.

Syllabus Topic : Digital Image Representation (2D Format, Resolution)

2.3.1 Digital Image Representation

Images can be in grayscale or in color. It is easier to understand the representation of grayscale images. Thus we first describe their representation and then extend it to color image representation.

Concept of Bits Per Pixel

Bpp or bits per pixel denotes the number of bits per pixel.
The number of different colors in an image is depends on the depth of color or bits per pixel.
Bits in mathematics:
Its just like playing with binary bits.
How many numbers can be represented by one bit.
0
1
How many two bits combinations can be made.
00
01
10
11
If we devise a formula for the calculation of total number of combinations that can be made from bit, it would be like this.

$$(2)^{bpp}$$

Where bpp denotes bits per pixel. Put 1 in the formula you get 2, put 2 in the formula, you get 4. It grows exponentially.

Number of different colors:

Now as we said it in the beginning, that the number of different colors depend on the number of bits per pixel.
The table for some of the bits and their color is given below.
This table shows different bits per pixel and the amount of color they contain.

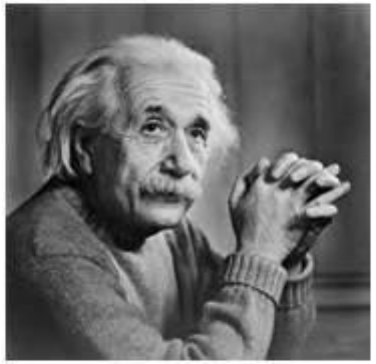
Bits per pixel	Number of colors
1 bpp	2 colors
2 bpp	4 colors
3 bpp	8 colors
4 bpp	16 colors
5 bpp	32 colors
6 bpp	64 colors
7 bpp	128 colors
8 bpp	256 colors
10 bpp	1024 colors
16 bpp	65536 colors
24 bpp	16777216 colors (16.7 million colors)
32 bpp	4294967296 colors (4294 million colors)

Gray color:
When you calculate the black and white color value, then you can calculate the pixel value of gray color.
Gray color is actually the mid point of black and white. That said,

In case of 8bpp, the pixel value that denotes gray color is 127 or 128bpp (if you count from 1, not from 0).

Image storage requirements
After the discussion of bits per pixel, now we have every thing that we need to calculate a size of an image.
Image size
The size of an image depends upon three things.
- Number of rows
- Number of columns
- Number of bits per pixel

The formula for calculating the size is given below.
Size of an image = rows * cols * bpp



It means that if you have an image, lets say this one.

Assuming it has 1024 rows and it has 1024 columns. And since it is a gray scale image, it has 256 different shades of gray or it has bits per pixel.
Then putting these values in the formula, we get
Size of an image = rows * cols * bpp
= 1024 * 1024 * 8
= 8388608 bits.

But since its not a standard answer that we recognize, so will convert it into our format.
Converting it into bytes = 8388608 / 8 = 1048576 bytes.
Converting into kilo bytes = 1048576 / 1024 = 1024kb.
Converting into Mega bytes = 1024 / 1024 = 1 Mb.

Thats how an image size is calculated and it is stored. Now in the formula,
if you are given the size of image and the bits per pixel, you can also calculate the rows and columns of the image, provided the image is square(same rows and same column).

Types of Images

There are many type of images, and we will look in detail about different types of images, and the color distribution in them.

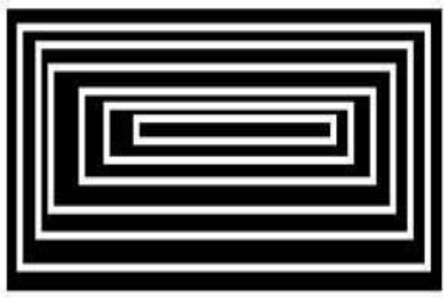
The binary image

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

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

Black and white image:
The resulting image that is formed hence consist of only black and white color and thus can also be called as Black and White image.

No 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.



Format

Binary images have a format of PBM (Portable bit map)

2, 3, 4,5, 6 bit color format

The images with a color format of 2, 3, 4, 5 and 6 bit are not widely used today. They were used in old times for old TV displays, or monitor displays.

But each of these colors have more then two gray levels, and hence has gray color unlike the binary image.

In a 2 bit 4, in a 3 bit 8, in a 4 bit 16, in a 5 bit 32, in a 6 bit 64 different colors are present.

8 bit color format

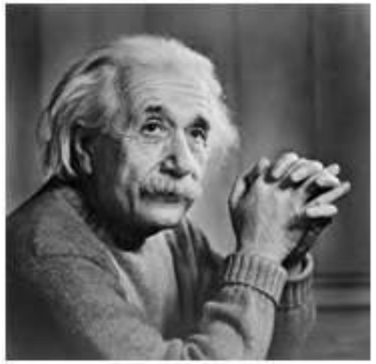
8 bit color format is one of the most famous image format. It has 256 different shades of colors in it.

It is commonly known as Grayscale image.

The range of the colors in 8 bit vary from 0-255. Where 0 stands for black,
and 255 stands for white, and 127 stands for gray color.

This format was used initially by early models of the operating systems UNIX and the early color Macintoshes.

A grayscale image of Einstein is shown below:



Format

The format of these images are PGM (Portable Gray Map).
This format is not supported by default from windows.
In order to see gray scale image, you need to have an image viewer or image processing toolbox such as Matlab.

16 bit color format

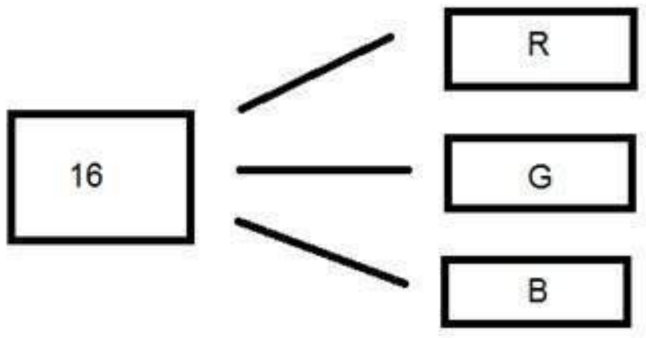
It is a color image format. It has 65,536 different colors in it. It is also known as High color format.

It has been used by Microsoft in their systems that support more then 8 bit color format.

Now in this 16 bit format and the next format we are going to discuss which is a 24 bit format are both color format.

The distribution of color in a color image is not as simple as it was in grayscale image.

A 16 bit format is actually divided into three further formats which are Red , Green and Blue. The famous (RGB) format.
It is pictorially represented in the image below.



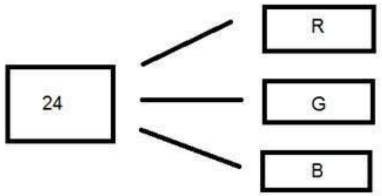
24 bit color format

24 bit color format also known as true color format. Like 16 bit color format, in a 24 bit color format, the 24 bits are again distributed in three different formats of Red, Green and Blue.

Since 24 is equally divided on 8, so it has been distributed equally between three different color channels.

Their distribution is like this.
8 bits for R, 8 bits for G, 8 bits for B.

Behind a 24 bit image.
Unlike a 8 bit gray scale image, which has one matrix behind it, a 24 bit image has three different matrices of R, G, B.



How different color codes can be combined to make other colors, and how we can covert RGB color codes to hex and vice versa.
Different color codes
All the colors here are of the 24 bit format, that means each color has 8 bits of red, 8 bits of green, 8 bits of blue, in it.
Or we can say each color has three different portions. You just have to change the quantity of these three portions to make any color.
Binary color format

Color: Black

Image:

Decimal Code:
(0,0,0)

Explanation:
an 8-bit format, 0 refers to black.
So if we have to make a pure black color, we have to make all the three portion of R, G, B to 0.

Color: White

Image:

Decimal Code:
(255,255,255)

Explanation:
Since each portion of R, G, B is an 8 bit portion. So in 8-bit, the white color is formed by 255.
In order to make a white color we set each portion to 255 and that's how we got a white color.
By setting each of the value to 255, we get overall value of 255, that's make the color white.

RGB color model:

Color: Red

Image:

Decimal Code:
(255,0,0)

Explanation:
Since we need only red color, so we zero out the rest of the two portions which are green and blue, and we set the red portion to its maximum which is 255.

Color: Green

Image:

Decimal Code:
(0,255,0)

Explanation:
Since we need only green color, so we zero out the rest of the two portions which are red and blue, and we set the green portion to its maximum which is 255.

Color: Blue

Image:

Decimal Code:
(0,0,255)
Explanation:
Since we need only blue color, so we zero out the rest of the two portions which are red and green, and we set the blue portion to its maximum which is 255

Gray color:

Color: Gray

Image:

Decimal Code:
(128,128,128)
Explanation
As we have already defined in our tutorial of pixel, that gray color Is actually the mid point. In an 8-bit format, the mid point is 128 or 127.
In this case we choose 128. So we set each of the portion to its mid point which is 128, and that results in overall mid value and we got gray color.

CMYK color model:

CMYK is another color model where c stands for cyan, m stands for magenta, y stands for yellow, and k for black. CMYK model is commonly used in color printers in which there are two carters of color is used. One consist of CMY and other consist of black color.
The colors of CMY can also made from changing the quantity or portion of red, green and blue.

Color: Cyan

Image:

Decimal Code:
(0,255,255)
Explanation:
Cyan color is formed from the combination of two different colors which are Green and blue. So we set those two to maximum and we zero out the portion of red. And we get cyan color.

Color: Magenta

Image:

Decimal Code:
(255,0,255)
Explanation:
Magenta color is formed from the combination of two different colors which are Red and Blue. So we set those two to maximum and we zero out the portion of green. And we get magenta color.

Color: Yellow

Image:

Decimal Code:
(255,255,0)
Explanation:
Yellow color is formed from the combination of two different colors which are Red and Green. So we set those two to maximum and we zero out the portion of blue. And we get yellow color.

Types of Images

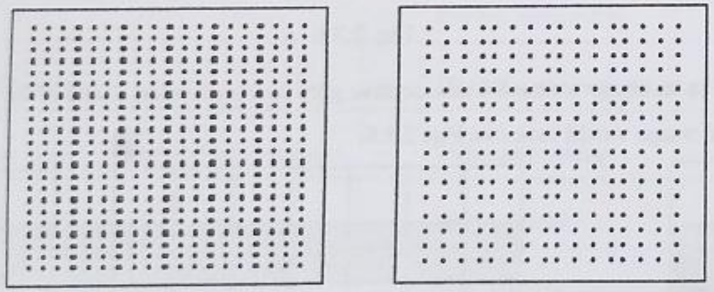
➜ (4) Half Toning

It is obvious that a grey scale image definitely looks better than the monochrome image as it utilizes more grey levels. But there is a problem in hand. Most of the printers that we use (inkjet, lasers, dot matrix) are all bi-level devices. i.e., they have only a black cartridge and can only produce two levels (black on a white back-ground). In fact, most of the printing jobs are done using bi-level devices.

You have all read newspapers at some point of time (hopefully). The images do look like grey level images. But if you look closely, all the images generated are basically using black colour. Refer Fig. 2.3.4.

Fig. 2.3.4

Even the images that you see in most of the books (including this one) are generated using black colour on a white background. In spite of this we do get an illusion of seeing grey levels. The technique to achieve an illusion of grey levels from only black and white levels is called half-toning.



The human eye integrates the scene that it sees. Consider a simple example. Consider two squares of say 0.03×0.03 sq.inch. One of these squares contains a lot of black dots while the other square contains fewer black dots. When we look at these squares from a distance, the two squares give us a perception of 2 different grey levels. This integration property of the eye is the basis for half toning. In this, we take a matrix of a fixed size and depending on the grey level required, we fill the matrix with black pixels.

Let us take an example.

Consider a 3×3 matrix. This matrix can generate an illusion of 10 different grey levels when viewed from a distance.

Here, the first block represents white, the last block represents black and all the other blocks represent intermediate values of grey. So in this case, while printing, if we encounter grey level 0, we plot 9 pixels which are all white. If we encounter grey level 1, we plot 9 pixels of which only one pixel is black and so on.

As is evident a 3×3 matrix will generate 10 different grey levels.

The dots that are placed in the 3×3 matrix example can be in any order. But we need to follow two rules.

(1) Dots should be arranged in such a manner so that they don't form straight lines. Lines are very obvious to the viewer and hence should be avoided.

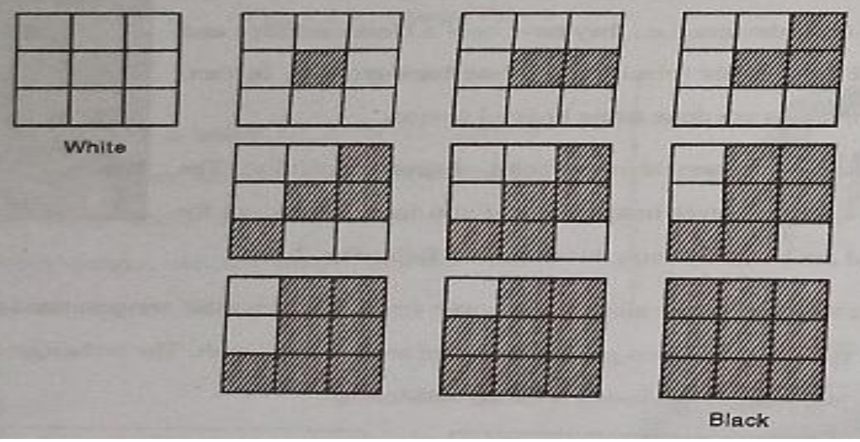


Fig. 2.3.6

Example, suppose in an image we have 4 consecutive grey levels of value 3, and if we use a code as sho
ig. 2.3.7, the half-toned image would look like Fig. 2.3.8.

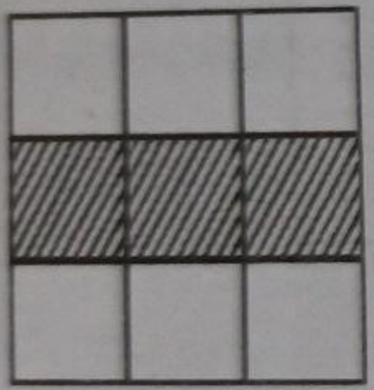


Fig. 2.3.7

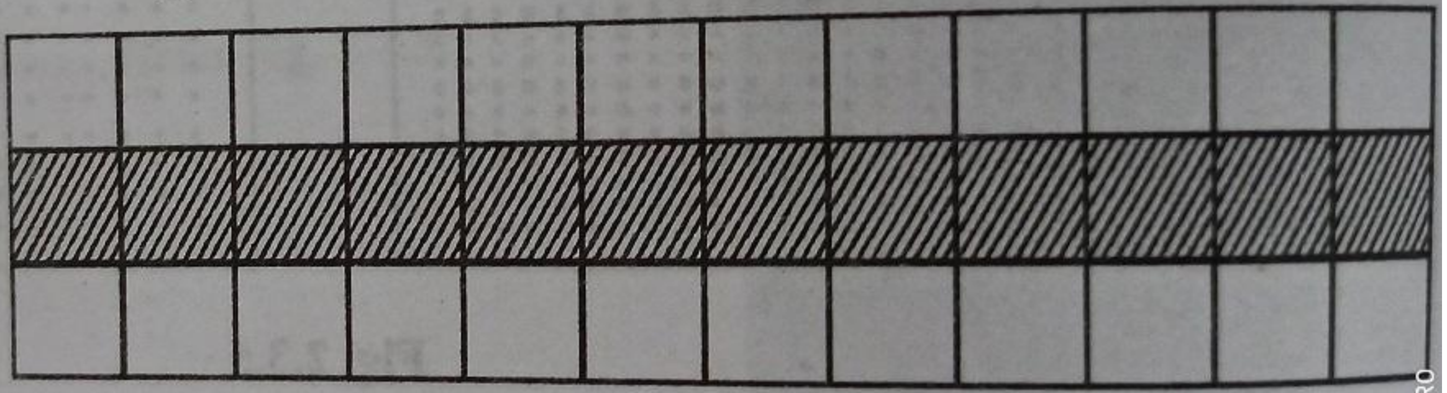


Fig. 2.3.8

It can be seen that it does not look like a grey level. It looks like a straight black line. Hence lines should be avoided while defining the matrices.

(2) If a pixel in a particular matrix is black for grey level i, it should be black for all further levels $j > i$. This reduces false contouring.

Half-toning gives excellent results and we do perceive a grey level image just by using black pixels on a white background.

The logic to implement a half-toned image from a grey level image is given below.

(1) Define the size of the half-toned matrices based on the number of grey levels the original image has.

(2) Generate the matrices starting from all white pixels to all black pixels.

(3) Read the original image. For every grey level value read, plot the corresponding matrix.

Remember, the physical size of the half-toned image will always be bigger than the original image as for every single grey level value, we output an entire matrix.

Examples of Images

2.3.3 Examples / Application of Images

The field of digital image processing has experienced continuous and significant expansion in recent years. The usefulness of this technology is apparent in many different disciplines covering medicine through remote sensing. The advances and wide availability of image processing hardware has further enhanced the usefulness of image processing.

– Medical applications	– Restorations and enhancements
– Digital cinema	– Image transmission and coding
– Color processing	– Remote sensing
– Robot vision	– Hybrid techniques
– Facsimile	– Pattern recognition
– Registration techniques	– Multidimensional image processing
– Image processing architectures and workstations	– Video processing
– Image sharpening and restoration	– Medical field
– Remote sensing	– Transmission and encoding
– Machine/Robot vision	– Color processing

Examples of Images

– Pattern recognition	– Video processing
– Microscopic imaging	– Programmable dsps for video coding
– High-resolution display	– High-quality color representation
– Super-high-definition image processing	– Impact of standardization on image processing.

☞ Image sharpening and restoration

Image sharpening and restoration refers here to process images that have been captured from the modern camera to make them a better image or to manipulate those images in way to achieve desired result. It refers to do what Photoshop usually does. This includes Zooming, blurring, sharpening, gray scale to color conversion, detecting edges and vice versa, Image retrieval and Image recognition.

☞ Medical field

The common applications of DIP in the field of medical is

- Gamma ray imaging
- PET scan
- X Ray Imaging
- Medical CT
- UV imaging

Example of Images

☞ UV imaging

– In the field of remote sensing, the area of the earth is scanned by a satellite or from a very high ground an then it is analyzed to obtain information about it. One particular application of digital image processing in th field of remote sensing is to detect infrastructure damages caused by an earthquake.

– As it takes longer time to grasp damage, even if serious damages are focused on. Since the area effected by th earthquake is sometimes so wide, that it not possible to examine it with human eye in order to estima damages. Even if it is, then it is very hectic and time consuming procedure. So a solution to this is found i digital image processing. An image of the affected area is captured from the above ground and then it i analyzed to detect the various types of damage done by the earthquake.

The key steps include in the analysis are extraction of edges and analysis and enhancement of various types o edges

1. Transmission and encoding

The very first image that has been transmitted over the wire was from London to New York via a submari cable. Now just imagine, that today we are able to see live video feed, or live cctv footage from one contine

Example of Images

to another with just a delay of seconds. It means that a lot of work has been done in this field too. This field does not only focus on transmission, but also on encoding. Many different formats have been developed for high or low bandwith to encode photos and then stream it over the internet or etc.

2. Machine/Robot vision

Apart from the many challenges that a robot face today, one of the biggest challenge still is to increase the vision of the robot. Make robot able to see things, identify them, and identify the hurdles etc. Much work has been contributed by this field and a complete other field of computer vision has been introduced to work on it.

3. Line follower robot

Most of the robots today work by following the line and thus are called line follower robots. This helps a robot to move on its path and perform some tasks. This has also been achieved through image processing.

Example of Images

4. **Color processing**

Color processing includes processing of colored images and different color spaces that are used. For example, RGB color model, YCbCr, HSV. It also involves studying transmission, storage and encoding of these color images.

5. **Pattern recognition**

Pattern recognition involves study from image processing and from various other fields that includes machine learning (a branch of artificial intelligence). In pattern recognition, image processing is used for identifying the objects in an images and then machine learning is used to train the system for the change in pattern. Pattern recognition is used in computer aided diagnosis, recognition of handwriting, recognition of images etc.

6. **Video processing**

A video is nothing but just the very fast movement of pictures. The quality of the video depends on the number of frames/pictures per minute and the quality of each frame being used. Video processing involves noise reduction, detail enhancement, motion detection, frame rate conversion, aspect ratio conversion, color space conversion etc.

Image File Formats

.4 Image File Formats

Images obtained from the camera are stored on the host computer using different formats. A file format is a tructure which defines how information is stored in the file and how that information is displayed on the monitor. There are numerous image file formats which are available. Of these only a few of them can be used universally. By universally it means they can be understood by different operating systems.

Some of the image formats that can be used on either Macintosh or PC platforms are;

BMP (Bit Mapped Graphic Image)	JPEG (Joint Photographic Expert Group)
TIFF (Tagged Image File Format)	EPS (Encapsulated Post Script)
GIF (Graphic Interchange Format)	PICT (Macintosh Supported)

TIFF, which is one of the most well known formats was developed in 1986 and in its many versions is a standard image format for a bit-mapped graphics image. The TIFF format is also a data compression technique for monochrome as well as colour images. Images seen on the Internet sites are normally TIFF/GIF images simply because they occupy less space. GIF uses a form of Huffman coding.

Image File Formats

BMP images developed by Microsoft can save both monochrome as well as colour images. All the wall papers that your computer has are BMP images. Similarly when we work in Paint Brush, we can save our work as a BMP image. The quality of BMP files is very good but they occupy a lot of memory space. PICT is a general purpose format supported by Macintosh and used for storing bit-mapped images. EPS is file format of the post script page description language and is device independent. This simply means that images can readily be transferred from one application to another. However EPS images can only be printed on post script compatible printers.

JPEG (Joint Photographic Expert Group) is the name of the committee that developed an image format which used compression algorithms.

JPEG images are compressed images which occupy very little space. It is based on the Discrete Cosine Transform-DCT. JPEG images use lossy compression algorithms which result in some loss of original data and hence the quality of JPEG images is not as good as BMP images.

Although these formats differ in technical details, they share structural similarities.

Fig. 2.4.1 shows the typical organisation of information encoded in an image file.

The image file consists of two parts :

(1) Header (2) Image data

The header file gives us the information about the kind of image. The header file, begins with a binary code or ASCII string which identifies the format being used. The width and height of the image are given in number of pixels. Common image types include binary images, 8-bit grey scale images, 8-bit colour and 24-bit colour images.

Image File Formats

The image data format specifies the order in which pixel values are stored in the image data section. A commonly used order is left to right and top to bottom. Image data format also specifies if the RGB values in the image are interlaced.

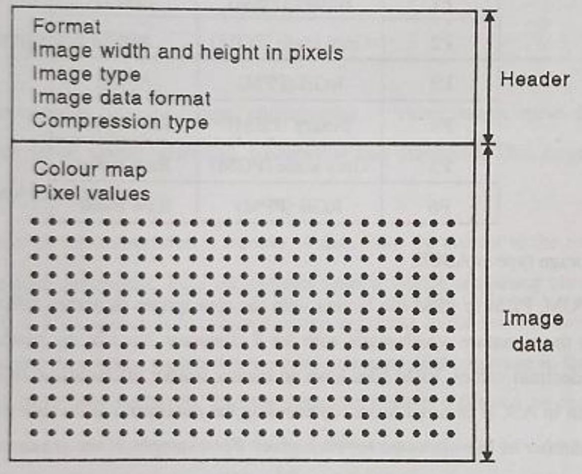


Fig. 2.4.1

By interlaced we mean that the RGB values of each pixel stay together consecutively followed by the three colour components for the next pixel i.e.,

$$\underbrace{R\ G\ B}_{1^{st}\ \text{Pixel}}\quad \underbrace{R\ G\ B}_{2^{nd}\ \text{Pixel}}\ \underbrace{R\ G\ B}_{\dots\dots\dots}\dots\dots$$

If the RGB values are not interlaced, the values of one primary colour for all pixels appear first, then the values of the next primary colour followed by the values of the third primary colour. Thus the image data is in the

Image File Formats

By interlaced we mean that the RGB values of each pixel stay together consecutively followed by the three colour components for the next pixel i.e.,

$$\underbrace{R\ G\ B}_{1^{st}\ \text{Pixel}}\ \ \underbrace{R\ G\ B}_{2^{nd}\ \text{Pixel}}\underbrace{R\ G\ B}_{\text{..........}}\ldots\ldots$$

If the RGB values are not interlaced, the values of one primary colour for all pixels appear first, then the values of the next primary colour followed by the values of the third primary colour. Thus the image data is in the form.

R R R R........ G G G G.......... B B B B.......

The compression type in the header indicates whether the image data is compressed using algorithms such as Run length encoding. Apart from these, there are a few more formats which we come across while dealing with images.

PGM (Portable Grey Map)	PGM is used to represent grey level images.
PBM (Portable Bit Map)	PBM is used to represent binary images
PPM (Portable Pixel Map)	PPM is used to represent RGB colour images

Image File Formats

These formats are distinguished by 2 character signatures as shown below.

Signature	Image type	Storage type
P1	Binary (PBM)	ASCII
P2	Grey scale (PGM)	ASCII
P3	RGB (PPM)	ASCII
P4	Binary (PBM)	Raw bytes
P5	Grey scale (PGM)	Raw bytes
P6	RGB (PPM)	Raw bytes

Most often the storage type is ASCII.

The header of a PGM, PBM or PPM file begins with the appropriate signature followed by a blank line. The is a comment line after the signature which starts with the # character. Next in the header are the width and heig of the image as ASCII decimal values. PBM files have no further header information. PGM and PPM files contai third integer value, again in ASCII decimal form, representing the maximum allowable pixel value. From this val we could find out the number of bits allocated for each pixel. For example, if the maximum allowable pixel value 25 each bit is represented by 8 bits ($2^8 = 256$).

Image File Formats

Fig. 2.4.2 shows a 8 × 8 grey scale image and its representation as a ASCII PGM file

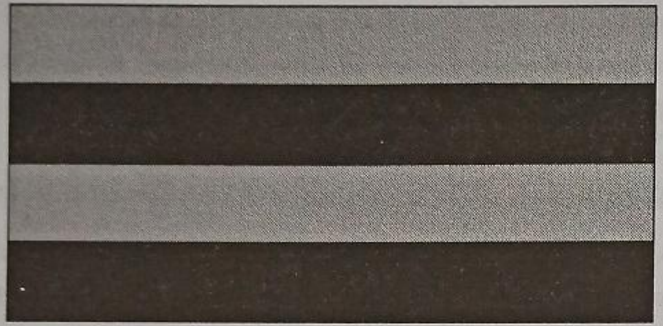


Fig. 2.4.2

Header

P2							
#A PGM image							
8	8	255					
120	120	120	120	120	120	120	120
120	120	120	120	120	120	120	120
33	33	33	33	33	33	33	33
33	33	33	33	33	33	33	33
120	120	120	120	120	120	120	120
120	120	120	120	120	120	120	120
33	33	33	33	33	33	33	33
33	33	33	33	33	33	33	33

Image File Formats

Just by observing the header, we know that it is a PGM file. P2 indicates that it is a grey level image stored ASCII. The $8 \times 8 \times 255$ below the comment line indicates that the size of the image is 8×8 and can have 256 gr levels i.e., each value is represented by 8-bits.

2.5 Data Compression

Multimedia object consists of color image, photographic or video image, audio data and full motion vide These data object needs to be stored, retrieved, transmitted and displayed. This large data objects present tv problems viz. storage and transmission.

Shannon-Fano Algorithm for Data Compression

DATA COMPRESSION AND ITS TYPES

Data Compression, also known as source coding, is the process of encoding or converting data in such a way that it consumes less memory space. Data compression reduces the number of resources required to store and transmit data.
It can be done in two ways- lossless compression and lossy compression. Lossy compression reduces the size of data by removing unnecessary information, while there is no data loss in lossless compression.

WHAT IS SHANNON FANO CODING?

Shannon Fano Algorithm is an entropy encoding technique for lossless data compression of multimedia. Named after Claude Shannon and Robert Fano, it assigns a code to each symbol based on their probabilities of occurrence. It is a variable length encoding scheme, that is, the codes assigned to the symbols will be of varying length.

HOW DOES IT WORK?

The steps of the algorithm are as follows:

1. Create a list of probabilities or frequency counts for the given set of symbols so that the relative frequency of occurrence of each symbol is known.

2. Sort the list of symbols in decreasing order of probability, the most probable ones to the left and least probable to the right.

3. Split the list into two parts, with the total probability of both the parts being as close to each other as possible.

4. Assign the value 0 to the left part and 1 to the right part.

5. Repeat the steps 3 and 4 for each part, until all the symbols are split into individual subgroups.

The Shannon codes are considered accurate if the code of each symbol is unique.

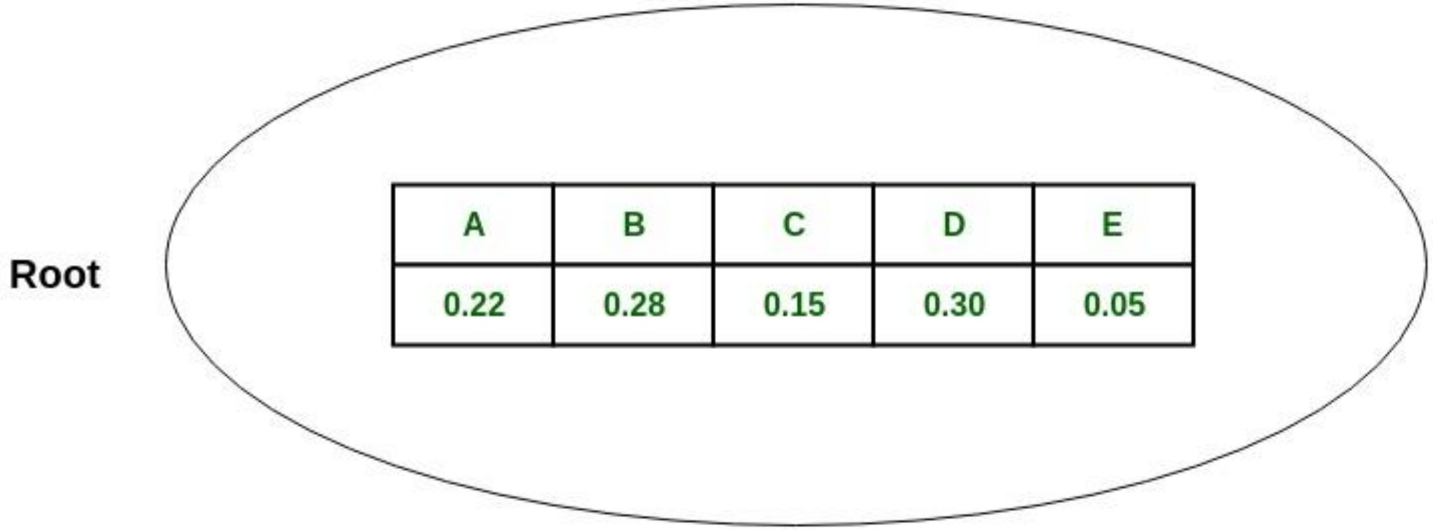
EXAMPLE:

Given task is to construct Shannon codes for the given set of symbols using the Shannon-Fano lossless compression technique.

Step

SYMBOL	A	B	C	D	E
PROBABILITY OR FRQUENCY	0.22	0.28	0.15	0.30	0.05

THE SYMBOLS AND THEIR PROBABILITY / FREQUENCY ARE TAKEN AS INPUTS.
(In case of Frequency, the values can be any number)



TREE AFTER STEP 1

Solution:

Let P(x) be the probability of occurrence of symbol x:

1. Upon arranging the symbols in decreasing order of probability:

P(D) + P(B) = 0.30 + 0.2 = 0.58

and,

P(A) + P(C) + P(E) = 0.22 + 0.15 + 0.05 = 0.42

And since the almost equally split the table, the most is divided it the blockquote table isblockquotento

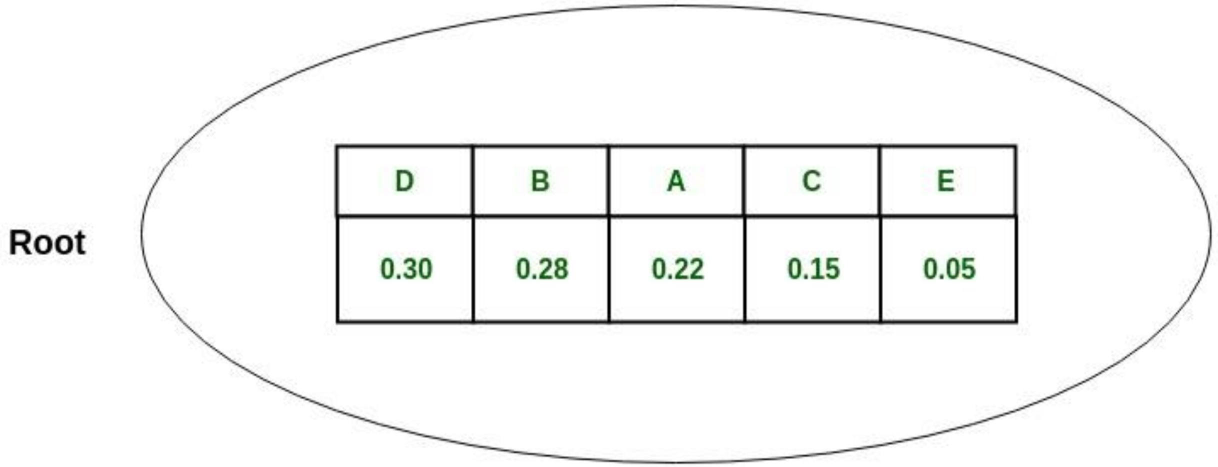
{D, B} and **{A, C, E}**
and assign them the values 0 and 1 respectively.

Step:

SYMBOL	D	B	A	C	E
PROBABILITY OR FRQUENCY	0.30	0.28	0.22	0.15	0.05

INPUTS ARE SORTED ACCORDING
TO THEIR PROBABILITY / FREQUENCY
(Here they are sorted according to their probability)

Tree:



TREE AFTER STEP 2

2. Now, in {D, B} group,

P(D) = 0.30 and **P(B) = 0.28**
which means that **P(D)~P(B)**, so divide {D, B} into {D} and {B} and assign 0 to D and 1 to B.
Step:

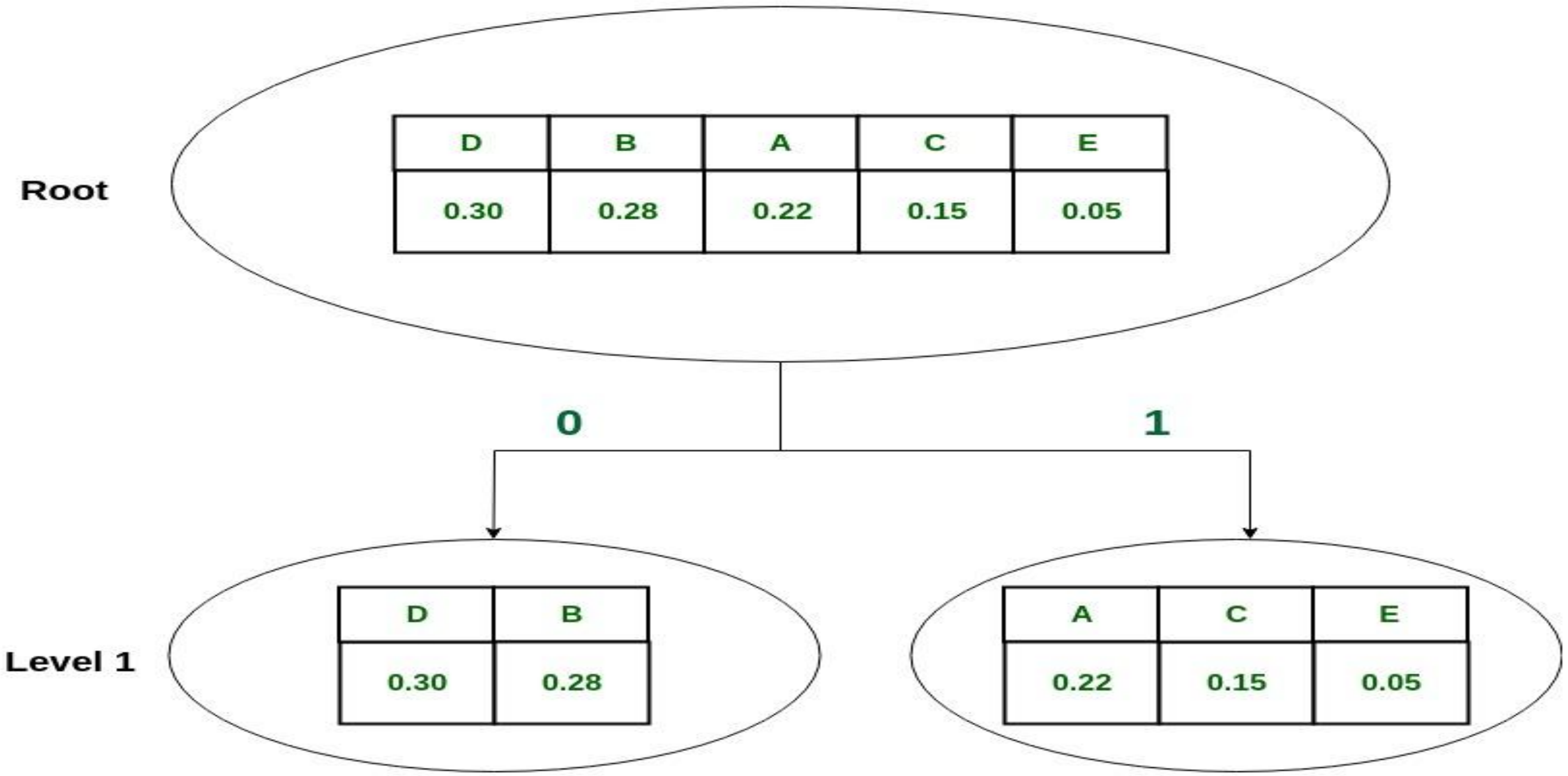
SYMBOL	D	B	A	C	E
PROBABILITY OR FRQUENCY	0.30	0.28	0.22	0.15	0.05

SYMBOL	D	B
PROBABILITY OR FRQUENCY	0.30	0.28

SYMBOL	A	C	E
PROBABILITY OR FRQUENCY	0.22	0.15	0.05

THE SYMBOLS ARE DIVIDED INTO TWO
SUCH THAT THE TOTAL PROBABILITY / FREQUENCY
OF LEFT SIDE ALMOST SAME AS THAT OF RIGHT SIDE

Tree:



TREE AFTER STEP 3

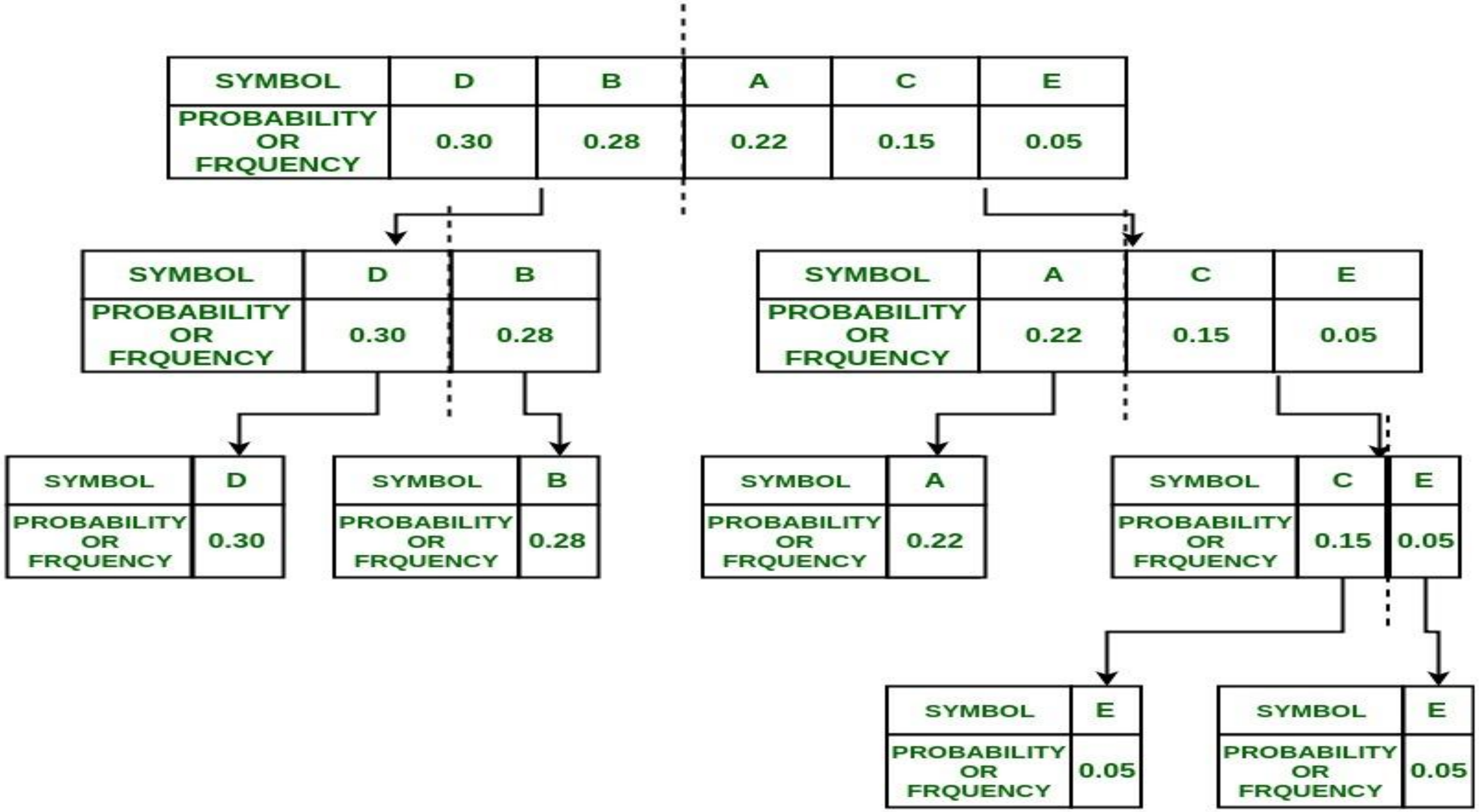
1. In {A, C, E} group,

P(A) = 0.22 and **P(C) + P(E) = 0.20**

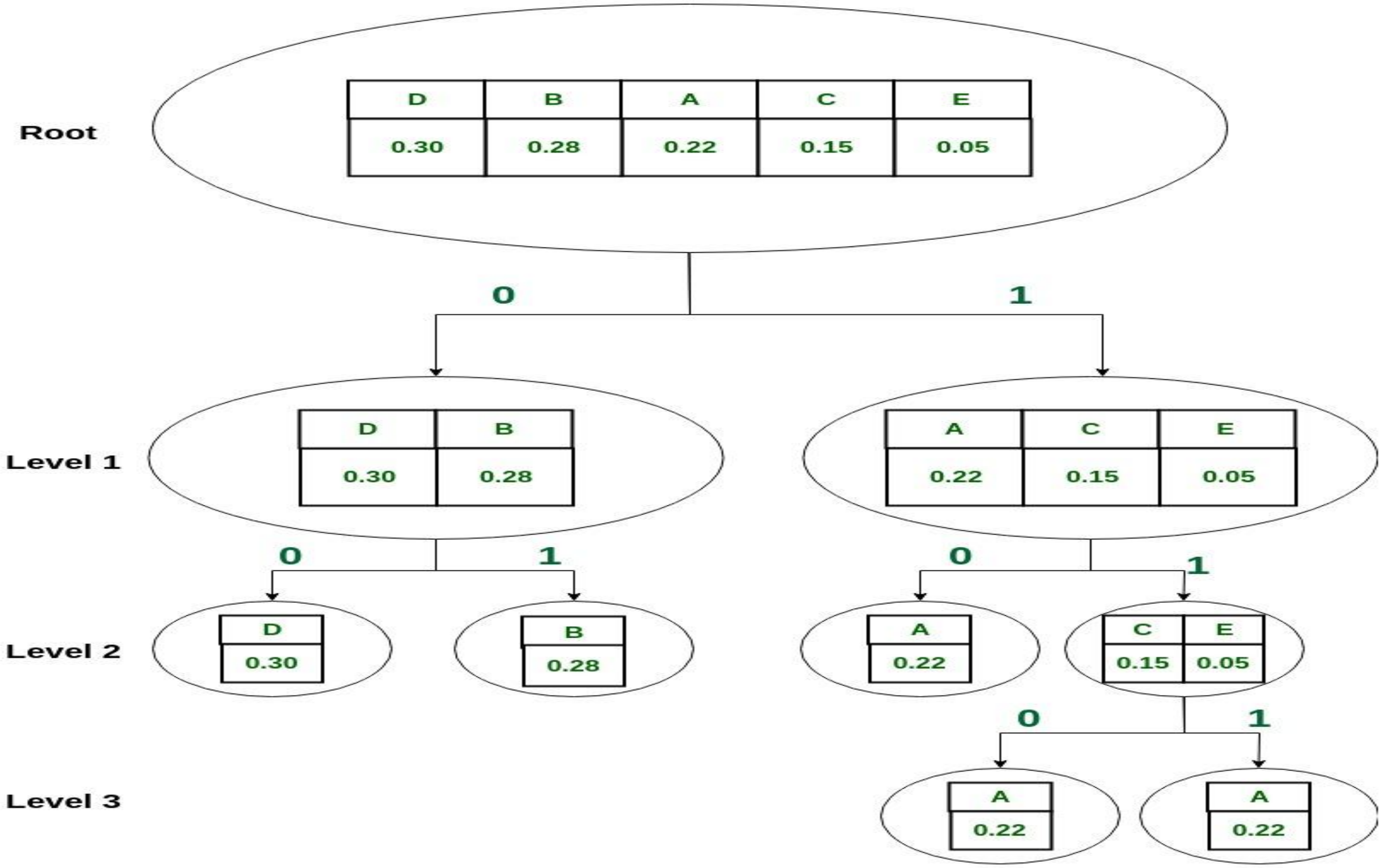
So the group is divided into
{A} and **{C, E}**
and they are assigned values 0 and 1 respectively.

2. In {C, E} group,

P(C) = 0.15 and **P(E) = 0.05**
So divide them into {C} and {E} and assign 0 to {C} and 1 to {E}



THE SYMBOLS ARE CONTINUED TO BE DIVIDED INTO TWO
TILL EACH SYMBOL BECOME SEPARATED



TREE AFTER STEP 4

Note: The splitting is now stopped as each symbol is separated now.

The Shannon codes for the set of symbols are:

SYMBOL	A	B	C	D	E
PROBABILITY OR FRQUENCY	0.22	0.28	0.15	0.30	0.05
SHANNON-FANO CODE	00	01	10	110	111

THE SHANNON CODES OF THE SYMBOLS
(based on their traversal from the root node)

As it can be seen, these are all unique and of varying lengths.