International Journal of Computer Science & Communication
Volume 6, Issue 2, April-September 2015
ISSN: 0973-7391 Impact Factor: 2.5 Available at www.csjournals.com

A Study of Various Data Compression Techniques

P.RAVI[1], Dr.A.ASHOKKUMAR[2]
Assistant Professor, Deptt of Comp. Sc., Government Arts College for Women, Tamilnadu,India
Assistant Professor, Deptt of Comp. Sc., Alagappa Government Arts College, Tamilnadu, India
peeravig@yahoo.co.in, ashokjuno@rediffmail.com

ABSTRACT: This paper proposed with the study of research work done in the field of various Data Compression. Image compression is the application of Data compression on digital images. Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and reduce communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications. The discrete cosine transform and wavelet transform are commonly used to reduce the redundancy between the pixels and for energy compaction. This paper entails the study of various image compression techniques and algorithms. In this paper we studied different types of data compression methods.
Keywords: Image Compression, Lossless, Lossy, Discrete Cosine Transform, wavelet transform

1. INTRODUCTION

Image compression is one of the application of Data compression on digital images. The objective of image compression is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression techniques are lossy and lossless.

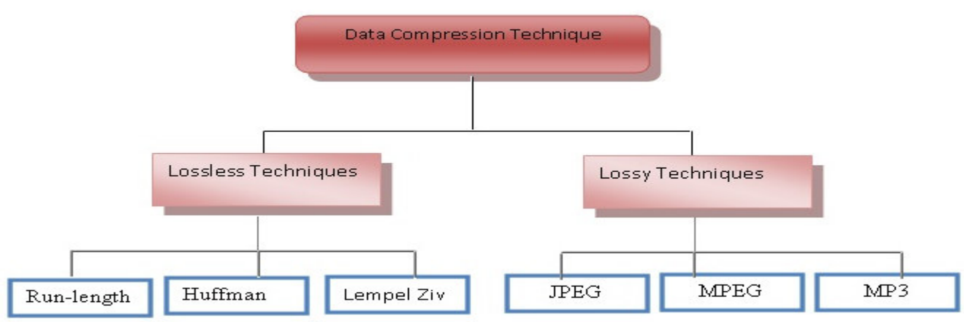

Fig 1: Data compression methods

Lossless compression is preferred for artificial images like technical drawings, icons and also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces unnoticeable differences can be called visually lossless. Run-length encoding, Huffman encoding and Lempel Ziv encoding are the methods for lossless image compression. Transform coding such as DCT ,Wavelet transform are applied followed by quantization and symbol coding can be cited as a method for lossy image compression. A general compression model is shown in fig 2. It shows that encoder and decoder consist of two relatively independent functions [2]. The encoder performs compression and the decoder performs the complementary operation of decompression. Both operation can be perform in software. Input image f(x,y) is fed into the encoder, which creates a compressed representation of the input. When the compressed representation is presented to its complementary decoder, a reconstructed output image $\hat{f}(x,y)$ is generated .In the encoding process a mapper transforms f(x,y) to reduce spatial and temporal redundancy and quantizer reduces the accuracy of the mapper 's output in accordance with pre-established fidelity criterion. The symbol coder to generates fixed or variable length code to represent the quantizer output. Symbol decoder and inverse mapper are perform the reverse operation of encoder's symbol coder and mapper respectively.

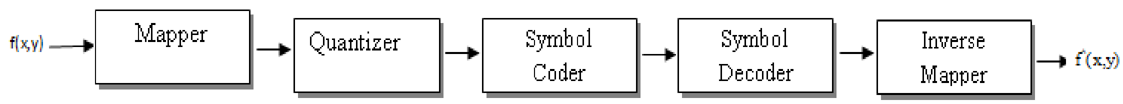

Fig 2: General Compression Model

2. LOSSLESS COMPRESSION

Lossless data compression is a class of data compression algorithms that allows the original data to be perfectly reconstructed from the compressed data. In lossless data compression, the integrity of the data is preserved. Redundant data is removed in compression and added during decompression. Lossless compression methods are normally used in cases where it is important that the original and the decompressed data be identical. There are few methods for Lossless compression given below:

2.1 Run-length encoding

Run-length encoding (RLE) is a very simple form of image compression in which runs of data are stored as a single data value and count, rather than as the original run. It is used for sequential [12] data and it is helpful for repetitive data. In this technique replaces sequences of identical symbol (pixel), called runs. The Run length code for a grayscale image is represented by a sequence { Xi , Yi } where Xi is the intensity of pixel and Yi refers to the number of consecutive pixels with the intensity Xi as shown in the figure. This is most useful on data that contains many such runs for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size. Run-length encoding performs lossless image compression [13]. Run-length encoding is used in fax coding.

Run-length encoding is particularly effective when compressing binary images. Because there are two possible intensities (black and white) ,adjacent pixel are more likely to be identical. It can be used to compress data made of any combination of two symbols represented as 0s and 1s.

The general idea behind this method is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences.

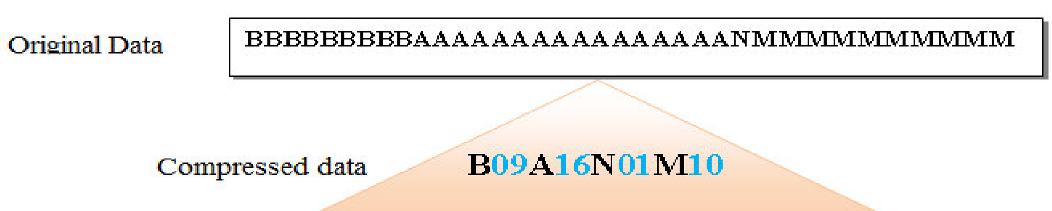

Fig 3: Run-length encoding compressed data.

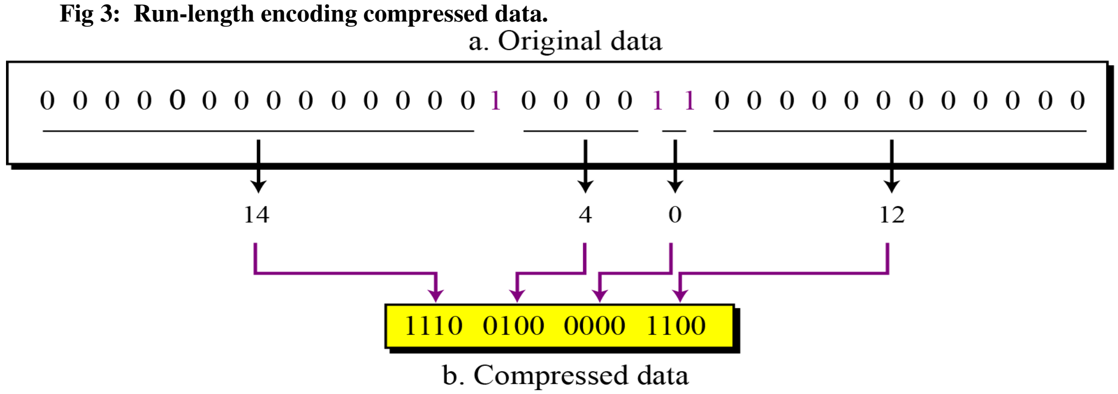

Fig 4: Run-length encoding for two symbols

2.2. Huffman coding

One of the most popular techniques for removing coding redundancy is due to Huffman. Huffman coding is an entropy encoding algorithm used for lossless data compression. It was developed by Huffman. Huffman coding [11] today is often used as a "back-end" to some other compression methods.

Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently. For example, we have a text file that uses only five characters (A, B, C, D, E). Before we can assign bit patterns to each character, we assign each character a weight based on its frequency of use. In this example, assume that the frequency of the characters is as shown in Table 1.

Table 1. Frequency of Characters

Character	**A**	**B**	**C**	**D**	**E**
Frequency	**17**	**12**	**12**	**27**	**32**

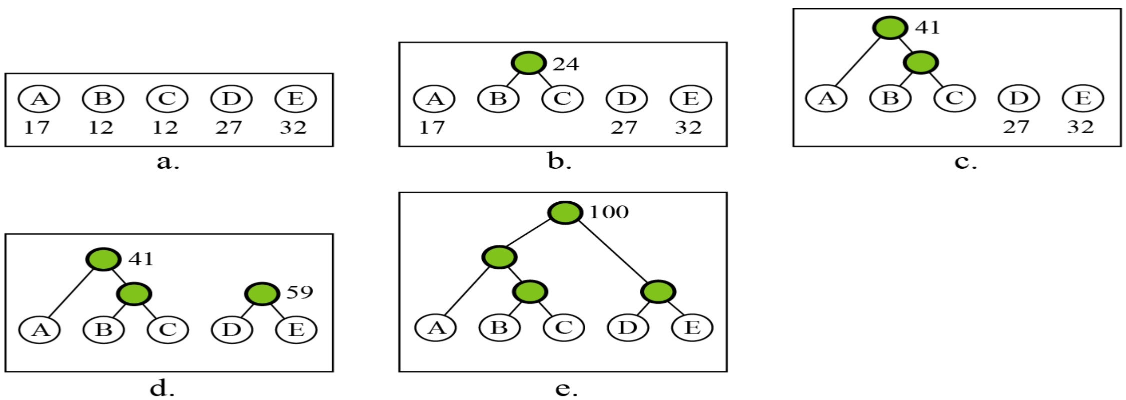

Fig 5:. Huffman coding tree

A character’s code is found by starting at the root and following the branches that lead to that character. The code itself is the bit value of each branch on the path, taken in sequence.

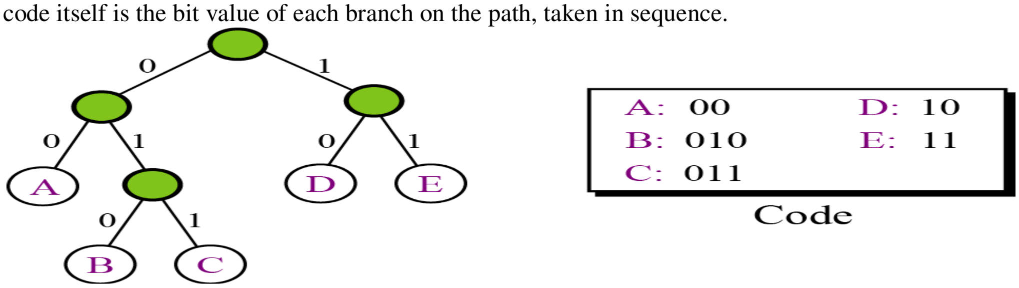

Fig 6: Final tree and code

1.2.1. Encoding : Let see how to implement encode text using the code for our five characters. Given Fig 7. shows the original text and the encoded symbol.

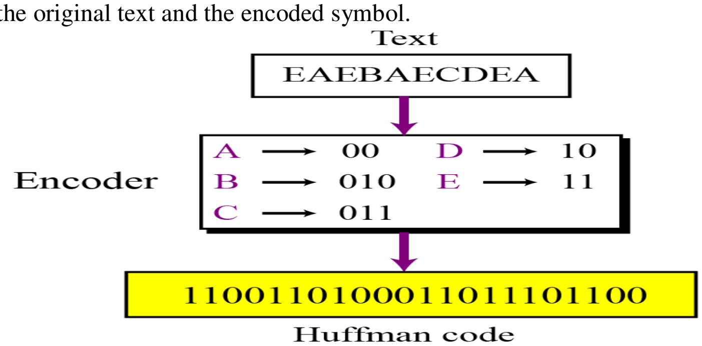

Fig 7: Huffman Encoding

2.2.2. Decoding:

The recipient has a very easy job in decoding the data it receives. In the Fig 8. show how to decoding the data from encoding symbol.

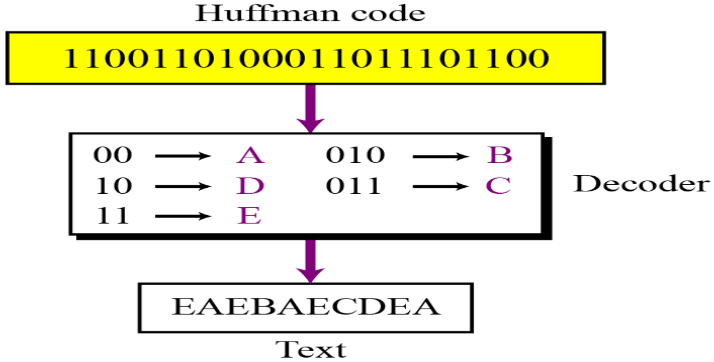

Fig 8: Huffman Decoding

2.3. Lempel Ziv encoding

Lempel–Ziv–Welch (LZW) is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZ78 algorithm published by Lempel and Ziv in 1978. LZW is a dictionary based coding. Dictionary based coding can be static or dynamic. In static dictionary coding, dictionary is fixed when the encoding and decoding processes. In dynamic dictionary coding, dictionary is updated on fly. The algorithm is simple to implement, and has the potential for very high throughput in hardware implementations. LZW compression has been integrated into a variety of mainstream imaging file format including GIF,TIFF and PNG. LZW compression became the first widely used universal image compression method on computers. A large English text file can typically be compressed via LZW to about half its original size.

Lempel Ziv (LZ) encoding is conceptually very simple. At the onset of the coding process, a dictionary The idea is to create a dictionary (a table) of strings used during the communication session. If both the sender and the receiver have a copy of the dictionary, then previously encountered strings can be substituted by their index in the dictionary to reduce the amount of information transmitted.

2.3.1 Compression

In this phase there are two synchronized events: building an indexed dictionary and compressing a string of symbols. The algorithm extracts the smallest substring that cannot be found in the dictionary from the remaining uncompressed string. It then stores a copy of this substring in the dictionary as a new entry and assigns it an index value. Compression occurs when the substring, except for the last character, is replaced with the index found in the dictionary. The process then inserts the index and the last character of the substring into the compressed string.

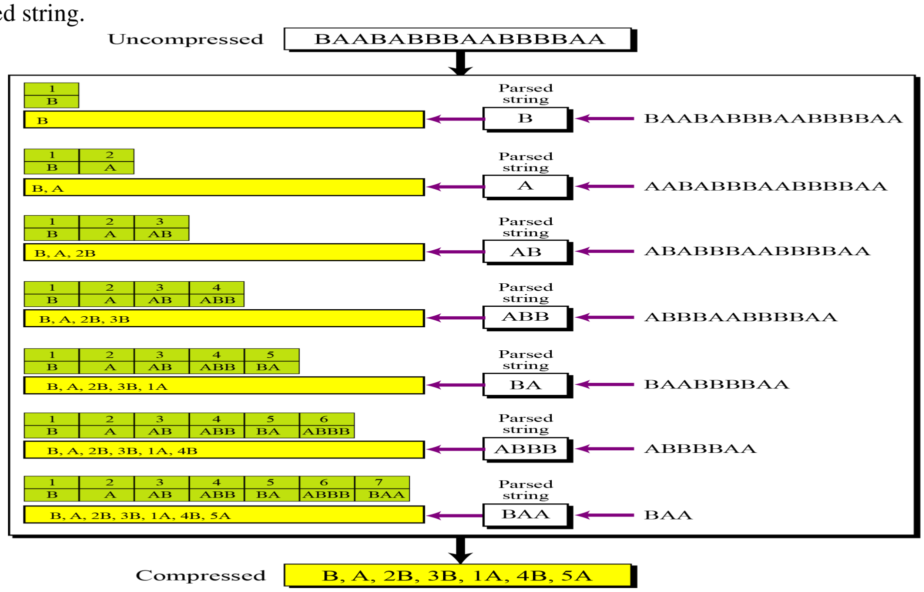

Fig 9: Lempel Ziv encoding

2.3.2 Decompression

Decompression is the opposite of the compression process. The process extracts the substrings from the compressed string and tries to replace the indexes with the corresponding entry in the dictionary, which is empty at first and built up gradually. The idea is that when an index is received, there is already an entry in the dictionary corresponding to that index.

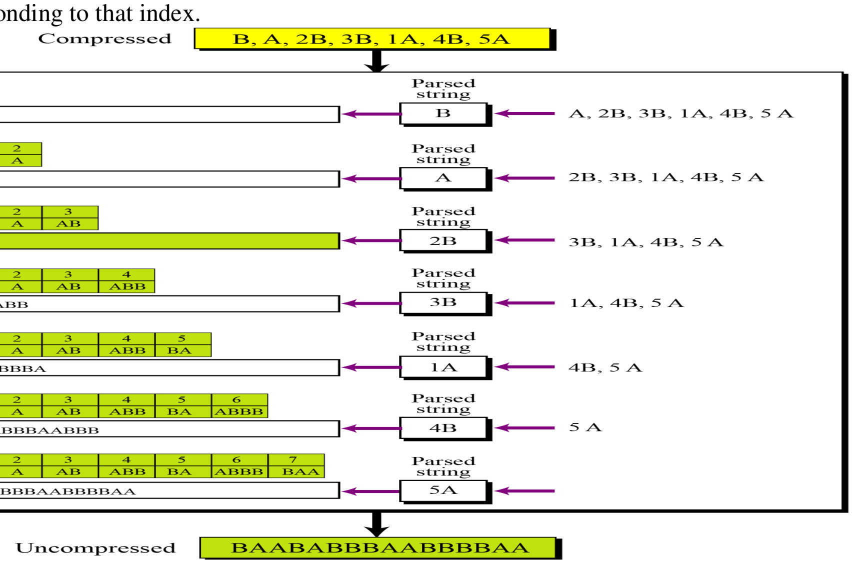

Fig 10: Lempel Ziv decoding

3. LOSSY COMPRESSION METHODS

Our eyes and ears cannot distinguish slight changes. In such cases, we can use a lossy data compression method. These methods are cheaper they take less time and space when it comes to sending millions of bits per second for images and video. Several methods have been developed using lossy compression techniques such as JPEG (Joint Photographic Experts Group) encoding is used to compress pictures and graphical images, MPEG (Motion of Picture Experts Group) encoding is used to compress video, and MP3 (MPEG audio layer 3) for digital audio compression.

3.1 Image compression JPEG encoding

An image can be represented by a two-dimensional array of picture elements. A grayscale image of 307,200 pixels is represented by 2,457,600 bits, and a color image is represented by 7,372,800 bits.

In JPEG, a grayscale picture is divided into blocks of 8 × 8 pixel blocks to decrease the number of calculations, because the number of mathematical operations for each picture blocks are the square of the number of units.

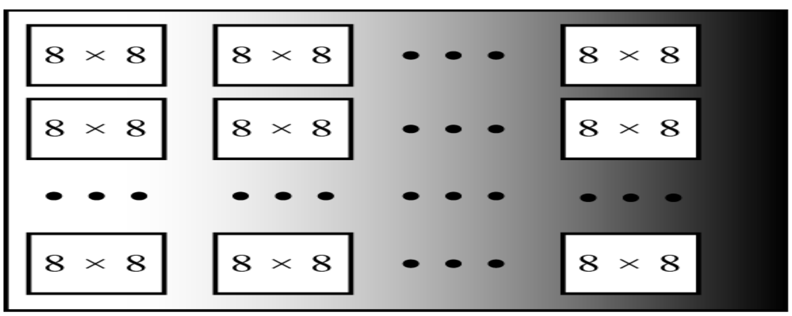

Fig 11: JPEG grayscale image, 640 × 480 pixels

JPEG is to change the picture into a linear set of numbers that reveals the redundancies. The redundancies can be removed using one of the lossless compression methods. A simplified version of the process is shown in Fig 12.

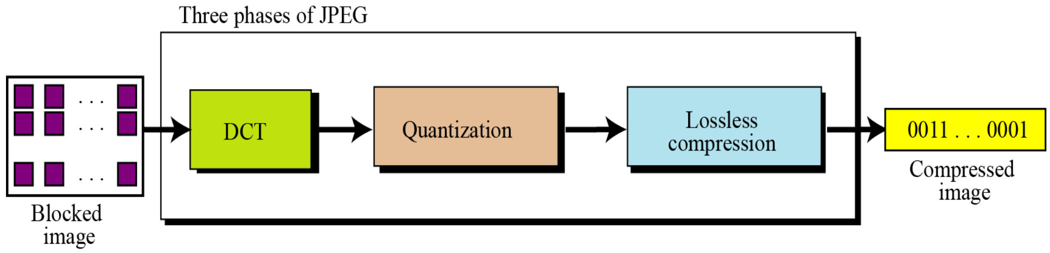

Fig 12: The JPEG compression process.

In this step, each block of 64 pixels goes through a transformation called the discrete cosine transform (DCT). The transformation changes the 64 (8x8) values so that the relative relationships between pixels are kept but the redundancies are revealed. The formula is P(x, y) defines one value in the block, while T(m, n) defines the value in the transformed block. To understand the nature of this transformation, let us show the result of the transformations for given cases:

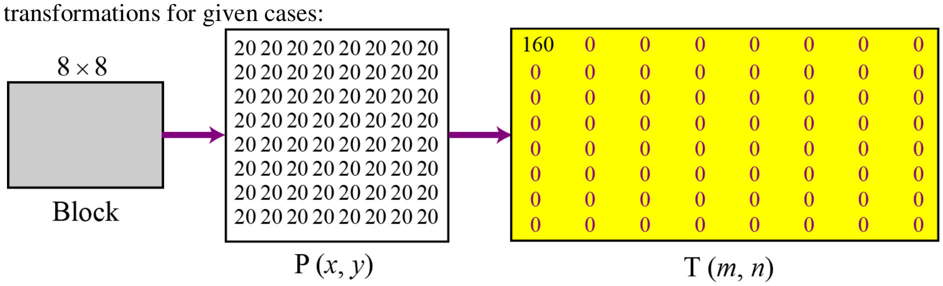

Fig 13: Case 1: Uniform grayscale

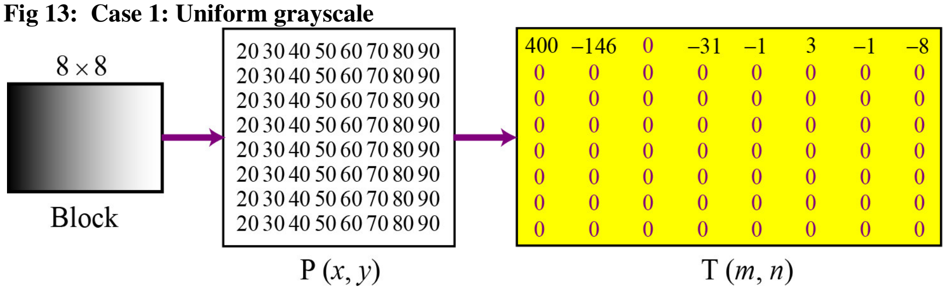

Fig 14: Case 2: Gradient grayscale

After the T table is created, the values are quantized to reduce the number of bits needed for encoding. Quantization divides the number of bits by a constant and then drops the fraction. This reduces the required number of bits even more. In most implementations, a quantizing table (8 x 8) defines how to quantize each value. The divisor depends on the position of the value in the T table. This is done to optimize the number of bits and the number of 0s for each particular application.

After quantization the values are read from the table, and redundant 0s are removed. However, to cluster the 0s together, the process reads the table diagonally in a zigzag fashion rather than row by row or column by column. The reason is that if the picture does not have fine changes, the bottom right corner of the T table is all 0s.

JPEG generally uses run-length encoding at the compression phase to compress the bit pattern resulting from the zigzag linearization.

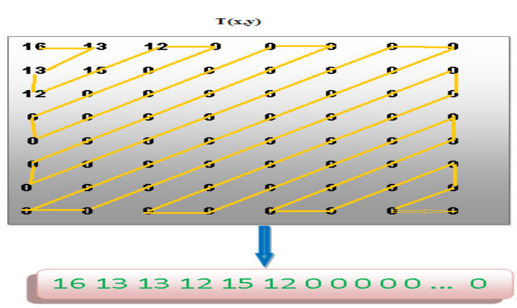

Fig 15: Reading the table

3.2 Video compression : MPEG encoding

The Moving Picture Experts Group (MPEG) method is used to compress video. In principle, a motion picture is a rapid sequence of a set of frames in which each frame is a picture. In other words, a frame is a spatial combination of pixels, and a video is a temporal combination of frames that are sent one after another. Compressing video, then, means spatially compressing each frame and temporally compressing a set of frames.

In spatial compression each frame is done with JPEG, or a modification of it. Each frame is a picture that can be independently compressed.

In temporal compression, redundant frames are removed. We receive 30 frames per second for single static video. However, most of the consecutive frames are almost the same. For example, in a static scene in which someone is talking, most frames are the same except for the segment around the speaker's lips, which changes from one frame to the next.

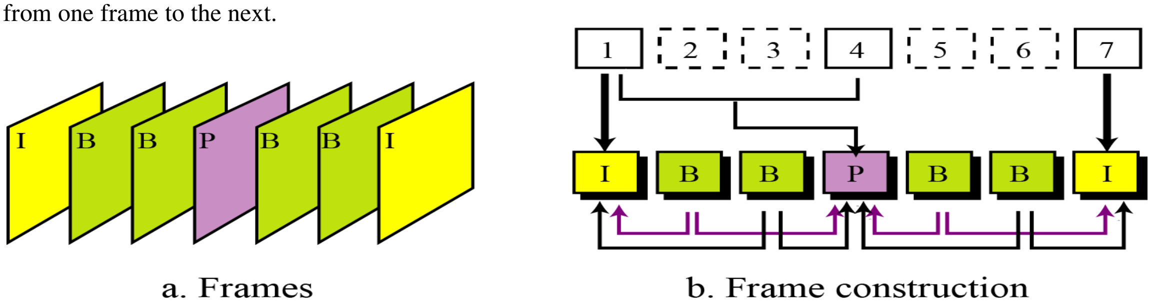

Fig 16: MPEG Frame Sequence

3.3 Audio Compression

Audio compression can be used for speech or music. For speech we need to compress a 64 kHz digitized signal, while for music we need to compress a 1.411 MHz signal. Two categories of techniques are used for audio compression: Linear predictive coding and perceptual coding.

3.3.1 Linear Predictive coding

Linear Predictive Coding is an analysis/synthesis technique to lossy speech compression that attempts to model the human production of sound instead of transmitting an estimate of the sound wave. Linear Predictive coding achieves a bit rate of 2400 bits/second which makes it ideal for use in secure telephone systems. Secure telephone systems require a low bit rate since speech is first digitalized, then encrypted and transmitted. These systems have a primary goal of decreasing the bit rate as much as possible while maintaining a level of speech quality that is understandable[11].

Telephone Systems	64 kb/s (uncompressed)
International Telephone Network	32 kb/s (can range from 5.3-64 kb/s)
Digital Cellular standards	6.7-13 kb/s
Regional Cellular standards	3.45-13 kb/s
Secure Telephony	0.8-16 kb/s

Fig 17 : Bit Rates for different telephone standards

In Linear predictive encoding, the differences between samples are encoded instead of encoding all the sampled values. This type of compression is normally used for speech. Several standards have been defined such as GSM (13 kbps), G.729 (8 kbps), and G.723.3 (6.4 or 5.3 kbps).

3.3.2 Perceptual coding: MP3

Digital coding at high bit rates is dominantly waveform-preserving, i.e., the amplitude versus time waveform of the decoded signal approximates that of the input signal. However, at lower bit rates, facts about the production and perception of speech and audio signals have to be included in coder design, and the error criterion has to be in favor of an output signal that is useful to the human receiver rather than favoring an output signal that follows and preserves the input waveform. Basically, an efficient source coding algorithm will (i) remove redundant components of the source signal by exploiting correlations between its samples and (ii) remove components which are irrelevant to the ear. Irrelevancy manifests itself as unnecessary amplitude or frequency resolution; portions of the source signal which are masked need not to be transmitted.

The encoding process is controlled by the signal- to-mask ratio (SMR) vs. frequency curve from which the necessary amplitude resolution (bit allocation and rate) in each critical band is derived. The SMR is the ratio of the sound pressures of signal and its masking threshold within a given frequency band. It is determined from a high resolution, say a 1024-point FFT-based spectral analysis of the audio block to be coded. Mainly, any coding scheme can be used that allows for a dynamic control by such perceptual information.
The most common compression technique used to create CD-quality audio is based on the perceptual encoding technique. This type of audio needs at least 1.411 Mbps, which cannot be sent over the Internet without compression. MP3 (MPEG audio layer 3) uses this technique.

4. CONCLUSION

This paper presents various techniques of data compression. These are still a challenging task for the researchers and academicians. There are mainly two types of data compression techniques exist. Comparing the performance of compression technique is difficult unless identical data sets and performance measures are used. Studied in different papers related to lossy and lossless compression techniques are used for better compression ratio for different types of data inputs. There are different types of symbol coding techniques that can be used for Data compression. After study of all techniques it is found that lossless image compression techniques are most effective over the lossy compression techniques.

5. REFERENCES:

[1] Comparative Analysis of Various Compression Methods for Medical Images : Rupinder . Kaur, Nisha Kaushal (NITTTR, Chd.) 2007

[2] Digital Image Processing,by Rafel C.Gonzalez and Richard E.Woods,Third Eition(2008)

[3] Hardware Implementation of a Lossless Image Compression Algorithm Using a Field Programmable Gate Array M. Klimesh,1 V. Stanton,1 and D. Watola1 (2001)

[4] Implementation of Data Compression and FFT on TinyOS: Ning Xu, Embedded Networks Laboratory, Computer Science Dept. USC. Los Angeles

[5] Implementation of Image Compression Algorithm using Verilog with Area, Power and Timing Constraints (2007-2009) : by Arun Kumar PS ,Dept. of ECE, NIT Rourkela.

[6] Image Compression Using the Discrete Cosine Transform By Andrew B. Watson, NASA Ames Research Center (1994)

[7] Data Compression by Behrouz Forouzan and Firouz Mosharraf, Foundation of computer Science .

[8] (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 10, October 2013 " A Survey: Various Techniques of Image Compression

[9]. K.Uma, **P.Geetha palanisamy***P.Geetha poornachandran "Comparison of Image Compression using GA, ACO and PSO techniques" IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011 978-1-4577-0590- 8/11 ©2011 IEEEMIT, Anna University, Chennai. June 3-5, 2011

[10] Linear Predictive Coding Jeremy Bradbury December 5, 2000

[11] Mridul Kumar Mathur, Seema Loonker and Dr. Dheeraj Saxena "Lossless Huffman Coding Technique For Image Compression And Reconstruction Using Binary Trees," IJCTA, pp.76-79, 2012

[12] Sonal, Dinesh Kumar," A Study of Various Image Compression Techniques," pp. 1-5.

[13] Tzong Jer Chen and Keh-Shih Chuang," A Pseudo Lossless Image Compression Method," EEE, pp. 610-615, 2010.

[14] Wei-Yi Wei," An Introduction to Image Compression", pp1-29.