INVESTIGATIONS ON HUFFMAN CODING FOR COMPRESSION OF BIT STREAM

Dishant Khosla[1], Sohni Singh[2], Anuj Kumar Gupta[3], Tejpal Sharma[4]
[1,4]Assistant Professor, [3]Professor, [2]Teaching Assistant
CGC College of Engineering, Mohali

Abstract. Compression techniques are nowadays becoming an eminent part in the era of computers due to wide growth in Multimedia and Internet. Assimilation of different types of computer data is the result of wide popularity of Multimedia. Multi-media combines many data types like graphics, text, animations, still images, audio and video data. Data Compression mrthods for different kinds of data do exist. Huffman Coding is a entropy or source of coding technique which tries to minimize the average length of messages in data and in turn reduces redundancy and also provides compression. This paper provides an overview of using the Huffman Coding in the compression of bit stream through the use of one or more other types of compression techniques.

Keywords: Source Coding, Compression, Entropy Coding, Redundancy.

I. INTRODUCTION

There has been a quick exchange of ideas and designs due to higher utilization of Multimedia and Internet. Numerous information such as sound, audio, still pictures etc. is consolidated within multimedia. There exists various kinds of data or wide range of compression techniques. The original data must be recovered in the exact form while using text compression techniques. The same applies to graphical data as well. Large storage is required for such type of data. In addition to the huge amount of storage, very large bandwidth is required otherwise there will be very less data transfer rate which is not at all desirable. If we wish to utilize our storage devices properly, compression is one of the tools that can be used which further uses not only one method but more than that. First we apply one technique and the result which is the encoded data is passed through another method of compression cycle. Large rates of compression can be accomplished [3]. In this paper, section II describes about the general compression model and the blocks that are necessary for data compression and their use. Section III describes how Huffman coding is used in compression of source symbols according to their probability of occurrence. Section IV gives an overview of how Huffman coding when used with any other compression technique affects compression of data.

II. GENERAL DATA COMPRESSION MODEL

A general compression model as shown in fig 1, shows that encoder and decoder consist of two relatively independent functions or sub blocks. The encoder is made up of source encoder, which removes input redundancies by using a number of techniques like Discrete Cosine transform, Quantizer and Huffman coding etc and a channel encoder, which increases the noise immunity of the source encoder's output by using a number of error detection and correction techniques like Hamming code and Cyclic Redundancy Check etc. Similarly, there is channel and source decoder. If there is no noise in the channel, then we do not require the decoder and encoder in the channel, so they result into source decoder and source encoder, respectively [1].

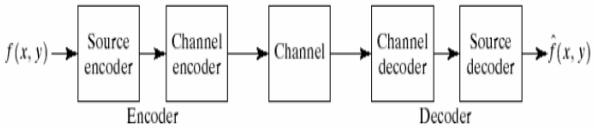

Fig 1. General Data Compression Model

A. Source Encoder:

Different steps are there to emit the redundancy when an encoder is designed. Firstly, a Mapper transforms input f(x, y) into a non visual format designed to reduce spatial and temporal redundancy. This operation is reversible and may or may not reduce directly the amount of data required to represent the image. Run length coding is an example of a mapping that normally yields compression in the first step of the encoding process. In video applications, the mapper uses previous video frames to facilitate the removal of temporal redundancy. Secondly, the Quantizer reduces the accuracy of mapper's output in accordance with a pre- established fidelity criterion. The data which is not required must be kept out of this type of method as it is not reversible and if we require compression which is completely free of error then this should be excluded. The encoded data bit rate is used if we wish to adjust quantizer so that we can maintain an output that is already determined in case of video applications. Thus, the output can be different in terms of image content. The length code is generated by symbol coder so that we can get the desired output. We can use a length code which is variable. We can reduce the redundancy in encoding if we assign the code which is the shortest to the output of quantizer which is

reversible. The redundancy is removed after the completion [1].

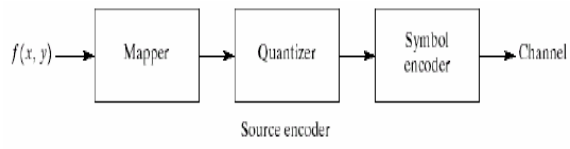

Fig 2: Source Encoder Model

B. Source Decoder:

The source decoder contains only two components which are a symbol decoder and an inverse mapper. They perform in reverse order, the inverse operations of the encoder's symbol encoder and mapper. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general decoder model. In video applications, decoded output frames are maintained in an internal frame store and used to reinsert the temporal redundancy that was removed at the encoder [1].

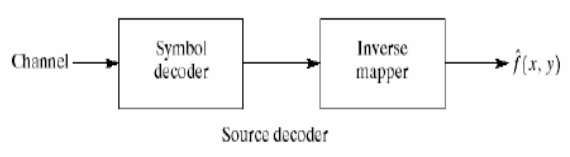

Fig 3: Source Decoder Model

III. HUFFMAN CODING ALGORITHM

One of the most popular techniques used for removing coding redundancy is Huffman coding. When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of Shannon's first theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time [4]. The source symbols may be either the intensities of an image or the output of an intensity mapping operation. Huffman coding is used to code values statically according to their occurrence. Short codes words are assigned to highly probable values and long code words to less probable values. Code generated by Huffman coding is instantaneous uniquely decodable block code [2].

Huffman codes are an important class of prefix codes. Every symbol in the alphabet is assigned to the data in the information that is needed to be sent. This is basically we call Huffman Coding. In the end we get a source code, the length of which is approximately the fundamental limit as in entropy, H [4]. The whole set is replaced with a simpler one in Huffman Coding. This process is continued unless we have two optimal code symbols which are 0 and 1. Now from backward we get the Huffman Code [2].

Specifically, the algorithm of Huffman encoding is:

- Initially, the symbols are written in the order of their decreasing probability.0 and 1 is given to the symbol which have the lowest probability. This is splitting stage.
- Now, a new symbol is created by combining these two symbols and its probability is sum of two probabilities. Now we can assign the probability of this new symbol with respect to its value.
- This is continued unless we are left with the symbols which are assigned only 0 and 1 [3].

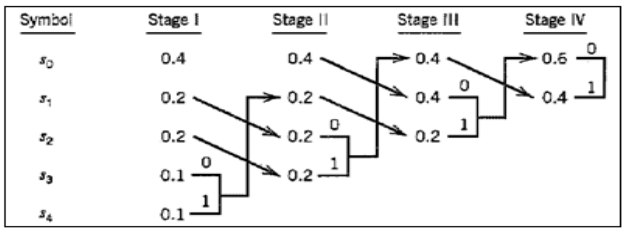

Table 1: Example of the Huffman Encoding algorithm.

Code word length (average) = 0.4 (2) + 0.2 (2) + 0.2 (2) + 0.1 (3) + 0.1 (3) = 2.2

Entropy H = $0.4 \log_2 (1 / 0.4) + 0.2 \log_2 (1 / 0.2) + 0.2 \log_2 (1 / 0.2) + 0.1 \log_2 (1 / 0.1) + 0.1 \log_2 (1 / 0.1)$ = 2.12193 bits

Symbol	Probability	Code word
s_0	0.4	00
s_1	0.2	10
s_2	0.2	11
s_3	0.1	010
s_4	0.1	011

Table 2: Source Code

IV. COMPRESSION USING HUFFMAN CODING

In this section, a brief overview is given on how Huffman Coding is used with some other compression techniques for image and video compression.

Lei (1990) et.al [5] discussed about implementation of an entropy coder and decoder for advanced TV applications. In this combining the two codings, an decoder and encoder of entropy are implemented for digital advanced television applications. Here, the outputs generated by the source coding algorithm have a high probability for the occurrence of zeros, especially in high frequency subbands. Thus, the RLC which represents a stream of continuous zeros by its length can effectively reduce the data rate. The VLC reduce the data rate further by assigning shorter codewords to the most frequently occurring source symbols. So in the decoder chip novel parallel architecture was used which decodes each codeword in one clock cycle regardless of the code length. Unlike the tree search, the parallel VLC decoder also has the flexibility of permitting a system design based on parallel paths to reduce the speed requirement further [5].

Lei (1991) et.al [6] proposed a full entropy coding in HDTV framework. In this there is coding and decoding without considering the length of the code. Thus, we can design a system with lower clock rate. Here RLC means continuous zeros so that samples can be minimized. The code words that occur frequently are assigned code words that are shorter by the VLC and in this way we can minimize the bit rate. RLC and VLC are able to compress the data successfully and deliver excellent compression effectiveness, practicable hardware application and minimum error. Here exact codewords can be initiated for length codeword organization and multiplexing. A proper technique of this is mentioned in the following [6].

Morimatsu (1995) et.al [7] developed a VLSI decoder chip fully compatible with MPEG 2 (Main Profile @ Main Level) video layer. The chip is also designed to work as a local decoder in a MPEG 2 real time encoder. The chip size and power dissipation are minimized by optimizing its architecture and by using macro cells for its bulky circuits such as multipliers. This chip has approximately 620K transistors on 11.35 x 11.35mm using a triple metal 0.5um CMOS technology [7].

Yang (2005) et.al [8] described the outline about Compression Techniques in Lossless Image. It has two steps: decorrelation as well as entropy coding. The spatial redundancy is minimizes in initial step or inter pixel, procedure that is dependent on SCAN, the methods which we can predict, the methods which can be transformed and various kinds of methods. The next step minimizes the redundancy with help of Huffman coding. The research is done on the first technique which is the decorrelation as we had discussed earlier. In this paper JPEG standard used for continuous tone image is overviewed with the algorithm that can be used in that [8].

Rigler (2007) et.al [9] devised LZ77 encoders as well as Huffman encoders which are perfectly suited for the application of GZIP. GZIP delivers better compression ratios. Here in the Huffman hardware implementation the GZIP encoder requires three powerful Huffman trees to pack a square of information. Every one of the three Huffman trees utilize similar structures yet the trees are of various sizes since each tree utilizes an alternate letters in order. The fundamental element of a dynamic Huffman encoder is that the recurrence information is computed for the info information and not read from a pre-characterized look into table. Some of RAM's are utilized for the capacity reason. One RAM is utilized to store the frequencies of each character. Two RAM's are utilized to store the qualities in the Huffman tree to monitor each hub's parent and profundity of every hub in the Huffman tree. Once the Huffman calculation is finished, the lengths of each character in the tree are figured and put away in another RAM. LZ equipment usage of the GZIP encoder comprises of 4 RAM's and a state machine. Two RAM's actualize a fastening hash table. Another RAM stores the info information. Another RAM stores separate length sets and literals to be composed as the yield. A number of variables are also used for the implementation [9].

Venugopal (2009) et.al [10] implemented Fast Variable Length Coder which is capable of processing very high resolution motion pictures of size 1600 x 1200 pixels at a frame rate of 30 per second meeting MPEG 2 standard. In this DCTQ and VLC are together used for compression. The video sequence that needs to be compressed is applied at the input of a Discrete Cosine Transform module. The DCT prepares the ground for compression by packing the picture information in as few coefficients as possible. This is followed by quantization, which succeeds in converting the DCT coefficients into many zeros which need not be coded in the next stage, the variable length coder. The VLC assigns shortest possible codes of varying lengths to the DCTQ coefficients, thus bringing about compression of the motion picture. The compression effected in VLC is much greater than that achieved during quantization. The VLC compression is lossless, whereas quantization is lossy. A brief description of various modules of the video encoder as implemented is given [10].

Ashok Babu (2010) et.al [11] implemented a proficient technique for interpreting the packed information through fast Huffman decoder. Here planning and execution of Huffman decoder for profoundly packed information was effectively done. The Huffman decoder utilizes a query table for recovering the first or transmitted information from the encoder. Interpreting of compacted information is done in view of correlation of every approaching piece with that of the accessible query table of code vectors. This query table comprises of all the remarkable words and their comparing code vectors. Once the Huffman coding is played out, the compacted information is transmitted serially to the decoder. At the decoder, this single piece information is acknowledged and in light of the way that the information can be either ' 0 ' or '1' its state is changed. After the event of the information, it is contrasted and every one of the code vectors frame the query table. On the off chance that the information matches with any of the code vectors, the decoder comprehends that the transmitter information from the encoder is the relating interesting expression of the coordinated code vector. In the event that the information does not coordinate with any of the code vectors from the query table, it acknowledges the following happening information and endeavors to coordinate with the code vector from the query table. Once a match of code vector with the got information design happens, the control is exchanged to the underlying state, showing that the information has been coordinated and the decoder is prepared to acknowledge the following piece. In the event that unmatched, control is exchanged to next state and again makes an endeavor to coordinate the query table. Comparable process proceeds till the total approaching information is decoded. This model enhanced the speed of decoding operation [11].

V. CONCLUSION

The exploding use of Multimedia and Internet has increased demand for speedy transfer of graphics and images. Compressing files before transmitting them saves telecommunications bandwidth. This requires selection of best compression method to be used so that maximum compression efficiency is obtained with minimum data loss. So this paper reviews different ways on how Huffman coding used with any other compression technique can increase compression efficiency.

REFERENCES

[1]. R. C. Gonzalez and R. E. Woods, "Digital Image Processing" Third Edition.

[2]. B. P. Lathi , " Modern Digital and Analog Communication System" Third Edition.

[3]. Simon Haykin , " Communication Systems " Fourth Edition.

[4]. David A Huffman "A Method for the Construction of Minimum Redundancy Codes" -1952.

[5]. Lei, Sun , Ramachandran and Palaniraj "VLSI Implementation of an Entropy Coder and Decoder for Advanced TV Applications" - 1990

[6]. Lei and Sun "An Entropy Coding System for Digital HDTV Applications" – March 1991.

[7]. Morimatsu, Sakai, Yamashita, Ohta, Miyasaka, Maeda, Ogura and Takeshita " Development of a VLSI Chip for Real Time MPEG 2 Video Decoder"- 1995.

[8]. Ming and Nikolaos "An Overview of lossless Digital Image Compression Techniques"- 2005.

[9]. Rigler, Bishop and Kennings "FPGA Based Lossless Data Compression using Huffman and LZ77 Algorithms"- 2007.

[10]. Venugopal , Ramachandran and Adiga "Design and FPGA Implementation of Fast Variable Length Coder for Video Encoder"- 2009.

[11]. Ashok and Satish "Implementation of Data Compression using Huffman Coding "- 2010.