Enhancing Efficiency of Huffman Coding using Lempel Ziv Coding for Image Compression

C. Saravanan, M. Surender

Abstract— ***Compression is a technology for reducing the quantity of data used to represent any content without excessively reducing the quality of the picture. The need for an efficient technique for compression of images ever increasing because the raw images need large amounts of disk space seems to be a big disadvantage during transmission & storage. Compression is a technique that makes storing easier for large amount of data. It also reduces the number of bits required to store and transmit digital media. In this paper, a fast lossless compression scheme is presented and named as HL which consists of two stages. In the first stage, a Huffman coding is used to compress the image. In the second stage all Huffman code words are concatenated together and then compressed with Lempel Ziv coding. This technique is simple in implementation and utilizes less memory. A software algorithm has been developed and implemented to compress and decompress the given image using Huffman coding techniques in MATLAB software.***

Keywords— ***Lossless image compression, Huffman coding, Lempel Ziv coding.***

I. INTRODUCTION

A digital image obtained by sampling and quantizing a continuous tone picture requires an enormous storage [3], [8], [9] . For instance, a 24 bit colour image with 512x512 pixels will occupy huge storage on a disk. To transmit such an image over a network would take more time. The purpose for image compression is to reduce the amount of data required for representing sampled digital images and therefore reduce the cost for storage and transmission [5]. Image compression plays a key role in many important applications, including image database, image communications, remote sensing, etc.

The image to be compressed is grayscale with pixel values between 0 and 255. Compression refers to reducing the quantity of data used to represent a file, image or video content without excessively reducing the quality of the original data. It also reduces the number of bits required to store and transmit digital media [7]. Compression could be defined as process of reducing the actual number of bits required to represent an image. There are different techniques to do that and they all have their own advantages and disadvantages. One of the methods commonly used for compression is to utilize the data redundancy. A redundant data may be represented using a simple technique which reduces the redundancy and hence the size also reduced. Another method is to find out region of interest in an image and apply one type of compression on the region of interest and other type of compression on other regions.

II. LITERATURE SURVEY

Yu-Ting Pai, Fan-Chieh Cheng, Shu-Ping Lu, and Shanq-Jang Ruan were proposed a technique in "Sub-Trees Modification of Huffman Coding for Stuffing Bits Reduction and Efficient NRZI Data Transmission". They mainly focused on the data transmission and multimedia compression and considered this problem as the encoding of compression and transmission to develop a low bit rate transmission scheme based on Huffman encoding [15]. The proposed method could balance "0" and "1" bits by analyzing the probability of the mismatch in the typical Huffman tree. Moreover, the proposed method also modified the transitional tree under the same compression ratio. Experimental results have showed that the proposed method could reduce the stuffing bits to 51.13% of standard JPEG compression.

In "A Hybrid Algorithm for Effective Lossless Compression of Video Display Frames" the authors Huang-Chih Kuo and Youn-Long Lin were proposed a simple and effective lossless compression algorithm for video display frames [17]. They combined the dictionary coding, the Huffman coding, and innovative scheme to achieve a high compression ratio. They quantitatively analyzed the characteristics of display frame data for designing the algorithm. First, a two-stage classification scheme to classify all pixels into three categories was proposed. Second, employ the dictionary coding and propose an adaptive prefix bit truncation scheme to generate codeword's for video pixels in each category. And subsequently employed the Huffman coding scheme to assign bit values to the codeword's. Finally, they proposed a head code compression scheme to further reduce the size of the codeword bits. Experimental results show that the proposed algorithm achieves 22% more reduction than prior arts.

Telemedicine characterized by transmission of medical data and images between users is one of the emerging fields in medicine. Huge bandwidth is necessary for transmitting medical images over the internet. So there is an immense need for efficient compression techniques to compress these medical images to reduce the storage space and efficiency of transfer the images over network for access to electronic patient records. Adina Arthur and V. Saravanan were proposed a method in "Efficient Medical Image Compression Technique for Telemedicine Considering Online and Offline Application". That method summarized the different transformation methods used in compression as compression is one of the techniques that reduced the amount of data needed for storage or transmission of information [14].

Manuscript received on January, 2013

Dr. C. Saravanan, Department of Computer Center, National Institute of Technology, Durgapur, West Bengal, India.

M. Surender, Department of Information Technology, National Institute of Technology, Durgapur, West Bengal, India.

This paper outlined the comparison of transformation methods such as DPCM (Differential Pulse Code Modulation) and prediction improved DPCM transformation step of compression then introduced a transformation which is efficient in both entropy reduction and computational complexity. A new method is then achieved by improving the prediction model which is used in lossless JPEG. The prediction improved transformation increased the energy compaction of prediction model and as a result reduced entropy value of transformed image. After transforming the image Huffman encoding used to compress the image.

Martin DVORAK and Martin SLANINA were presented a simple model of a video codec in “Educational Video Codec” which is used for educational purposes [10]. The model is based on the MPEG-2 video coding standard and is implemented in the MATLAB environment, which allows for easy optimization, modification or creation of new compression tools and procedures. The model contained the basic compression tools for video signal processing and allows extensive settings of these tools. These settings give users an opportunity to learn the basic principles of compression and they also provide an overview of the effectiveness of individual tools. The application is designed primarily for video compression with low resolution.

Data compression has been playing an important role in the areas of data transmission. Many great contributions have been made in this area, such as Huffman coding, LZW algorithm, run length coding, and so on. These methods only focus on the data compression. On the other hand, it is very important for us to encrypt our data to against malicious theft and attack during transmission. A novel algorithm named Swapped Huffman code Table (SHT algorithm) which has joined compression and encryption based on the Huffman coding. This technique was proposed in “Enhanced Huffman Coding with Encryption for Wireless Data Broadcasting System” by Kuo-Kun Tseng, JunMin Jiang, Jeng-Shyang Pan, Ling Ling Tang, Chih-Yu Hsu, and Chih-Cheng Chen [6].

From the above Literature survey we conclude that Huffman coding technique is very effectively used for compressing the images in a different way. A different technique proposed HL Scheme using Huffman coding and Lempel Ziv coding to compress the image and observed the experiment results. The proposed HL scheme enhances the efficiency of the Huffman coding technique.

III. PROPOSED HL SCHEME

This paper proposes a compression technique using the two lossless methodologies Huffman coding and Lempel Ziv coding to compress image. In the first stage, the image is compressed with Huffman coding resulting the Huffman tree and Huffman Code words. In the second stage, all Huffman code words are concatenated together and then compressed by using Lempel Ziv coding. Normally, the Huffman tree and Huffman code words are stored to recover the original image. In the proposed technique, the Huffman code words are applied compression using the Lempel Ziv coding to reduce size. From the experiment results it is observed that the size of the data reduced further. To recover the image data, first decompress the image data by Lempel Ziv decoding technique and as a second step apply Huffman decoding.

A. First step-Huffman coding

The Huffman code is designed by merging the lowest probable symbols and this process is repeated until only two probabilities of two compound symbols are left and thus a code tree is generated and Huffman codes are obtained from labeling of the code tree [7], [11].

For example the data to encode is “13371545155135706347”. By using Huffman coding technique the letters correspond to leaves and following the edges from top to bottom one can read off the code for a given letter. To construct the "optimal" tree for a given alphabet and given probabilities one follows the following algorithm [2]:

1. Create a list of nodes. Each node contains a letter and its probability.
2. Find the two nodes with lowest probabilities in the list.
3. Make them the children of a new node that has the sum of their probabilities as probability.
4. Remove these children from the list and add the new parent-node.
5. Repeat the last three steps until the list contains only one node.

The tree from our example was created following this algorithm with the probabilities chosen according to the number of occurrences in the string. Let consider the following table I represents source symbols and their probability as follows.

Table I: Symbols and Probability

Symbols	Probability
0	0.05
1	0.2
2	0
3	0.2
4	0.1
5	0.25
6	0.05
7	0.15

The dictionary which is equivalent to a binary tree for our example by using Huffman coding would be [13] as follows.

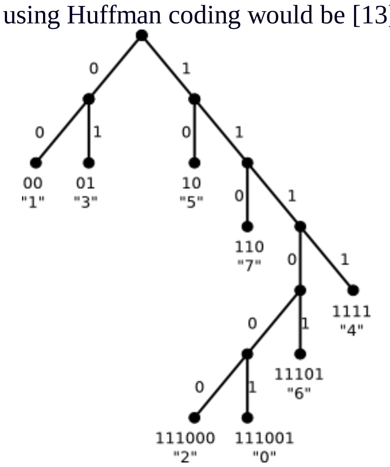

Fig 1: Huffman Tree

From the above figure 1 we get the Huffman code words for the respective symbols as follows [12].

Table II: Huffman codeword's

Symbol	Huffman code words
1	00
3	01
5	10
7	110
2	111000
0	111001
6	11101
4	1111

Suppose the message we want to encode is '1' '3' ' 3' '7' '1' '5' '4' '5' '1' '5' '5' '1' '3' '5' '7' '0' '6' '3' '4' '7'.

Having the table ready, the Huffman encoding itself is very simple. Encode this message by using Huffman tree then the resulting Huffman code words are

00,01,01,110,00,10,1111,10,00,10,10,00,01,10,110,111001, 11101,01,1111,110.

B. Second step-Lzw encoding

In the second step concatenate all the Huffman codeword's and apply the following Lzw encoding algorithm.

A high level view of the encoding algorithm is shown here [4], [1]:

1. Initialize the dictionary to contain all strings of length one.
2. Find the longest string W in the dictionary that matches the current input.
3. Emit the dictionary index for W to output and remove W from the input.
4. Add W followed by the next symbol in the input to the dictionary.
5. Go to Step 2.

The concatenated Huffman code words are 0001011100010111110001010000110110111001111010111111110 and the Lzw dictionary for above is represented as Table III

Table III: Lzw Dictionary

Dictionary Location	Alphabet	Encoded words
0	Φ	-
1	0	(0,0)
2	00	(1,0)
3	1	(0,1)
4	01	(1,1)
5	11	(3,1)
6	000	(2,0)
7	10	(3,0)
8	111	(5,1)
9	110	(5,0)
10	001	(2,1)
11	010	(4,0)
12	0001	(6,1)
13	101	(7,1)
14	1011	(13,1)
15	100	(7,0)
16	1111	(8,1)
17	0101	(11,1)
18	11111	(16,1)
19(dup)	10	(3,0)

and the final encoded values are

(0,0),(1,0),(0,1),(1,1),(3,1),(2,0),(3,0),(5,1),(5,0),(2,1),(4,0),(6,1),(7,1),(13,1),(7,0),(8,1),(11,1),(16,1),(3,0).

C. Lzw Decoding

The decoding algorithm works by reading a value from the encoded input and outputting the corresponding string from the initialized dictionary. At the same time it obtains the next value from the input, and adds to the dictionary the concatenation of the string just output and the first character of the string obtained by decoding the next input value. The decoder then proceeds to the next input value (which was already read in as the "next value" in the previous pass) and repeats the process until there is no more input, at which point the final input value is decoded without any more additions to the dictionary [4]. The result is the concatenated Huffman code words i.e.
0001011100010111110001010000110110111001111010111111110

D. Huffman decoding

After the code has been created, coding and/or decoding is accomplished in a simple look-up table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code, because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way [2]. Thus, any string of Huffman
encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner [16].

After decoding the above Huffman codeword by using the Huffman tree we get the original message i.e. "13371545155135706347".

IV. ALGORITHM AND FLOW CHART

The algorithm for proposed HL scheme is given as

Step1- Read the image on to the workspace of the mat lab.
Step2- Convert the given colour image into grey level image.
Step3- Call a function which will find the symbols (i.e. pixel value which is non-repeated).
Step4- Call a function which will calculate the probability of each symbol.
Step5- Probability of symbols are arranged in decreasing order and lower probabilities are merged and this step is continued until only two probabilities are left and codes are assigned according to rule that the highest probable symbol will have a shorter length code.
Step6- Further Huffman encoding is performed i.e. mapping of the code words to the corresponding symbols will result in a Huffman codeword's
Step7- Concatanate all the Huffman code words and apply Lzw encoding will results in Lzw Dictionary and final encoded Values (compressed data).
Step8- Apply Lzw decoding process on Final Encoded values and output the Huffman code words
Step9- Generate a tree equivalent to the encoding tree.
Step10-Read input character wise and left to the table until last element is reached in the table.
Step11-Output the character encodes in the leaf and returns to the root, and continues the table until all the codes of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

corresponding symbols are known.

Step12-The original image is reconstructed i.e. decompression is done by using Huffman decoding.

The flow chart for our proposed HL scheme as shown below.

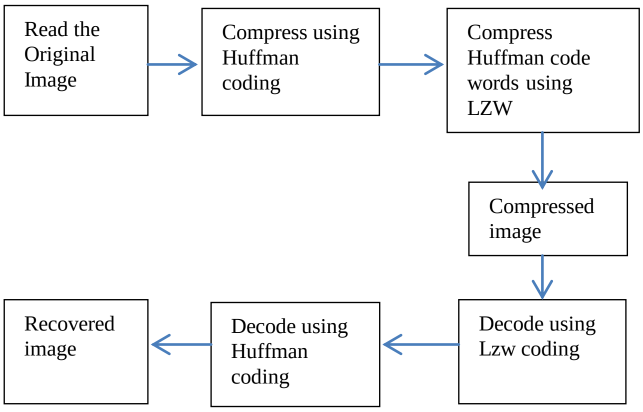

Fig 2: Flow chart

V. RESULTS

The experiment results are shown below.

Table IV: Compressed Size of image using Huffman coding and HL scheme

Image Name	Size in bytes	Compressed Size Using Huffman coding (HC)	Compressed Size using HL Scheme
Lena16	2048	1512	1224
Lena32	8192	6552	3808
Lena64	32768	23952	10136
Lena128	1,31072	94,974	28456

Table V: Compression Ratio of Huffman coding and HL scheme

Image Name	Compression Ratio using HC (in %)	Compression Ratio using HL Scheme (%)	Difference
Lena16	74	60	14
Lena32	80	46	34
Lena64	73	31	42
Lena128	72	22	50

From the above two tables IV and V, it is observed that the proposed compression scheme outperforms the Huffman coding and also improves the efficiency of the Huffman coding up to a great extent of 50 percentage compression ratio.

VI. CONCLUSION

This experiment confirms that the higher data redundancy helps to achieve more compression. The above presented new compression and decompression technique called as HL Scheme based on Huffman coding and Lempel Ziv coding is very efficient technique for compressing the image to a great extent, as observed from the results of the experiment from table 5, the proposed HL Scheme achieves at the most 50 percentage compression ratios to the Huffman coding. The HL Scheme also results in an algorithm with significant time, and advantage become most apparent for images with more in size. Sine, the reproduced image and the original image are equal; the HL Scheme is a lossless compression scheme. As a future work more focus shall be on improvement of compression ratio.

REFERENCES

1. http://www.TheLZWcompressionalgorithm.html
2. http://en.wikipedia.org/wiki/huffmancoding
3. http://en.wikipedia.org/wiki/Lossless_JPEG
4. J.Ziv and A.Lempel "A Universal Algorithm for Sequential data compression", IEEE Transaction on Information theory, May 1977.
5. Introduction to Data Compression, Khalid Sayood, Ed Fox (Editor), March 2000.
6. Kuo-Kun Tseng, JunMin Jiang, Jeng-Shyang Pan, Ling Ling Tang, Chih-Yu Hsu, and Chih-Cheng Chen, "Enhanced Huffman Coding with Encryption for Wireless Data Broadcasting System", International Symposium on Computer, Consumer and Control, 2012.
7. Mamta Sharma, "Compression Using Huffman Coding", International Journal of Computer Science and Network Security, Vol.10, No.5, May 2010.
8. C. Saravanan and R. Ponalagusamy, "Lossless Grey-scale Image Compression using Source Symbols Reduction and Huffman Coding", International Journal of Image Processing (CSC Journals), Vol.3, Iss.5, pp.246-251, 2009.
9. M.J.Weinberger, G.Seroussi, and G. Saprio. "LOCO-1 Lossless, Image Compression Algorithm: Principles and Standardization into JPEG-LS", IEEE trans. Image Processing, pp.1309 - 1324, August 2000.
10. Martin DVORAK, Martin SLANINA, "Educational Video Codec", 22nd International Conference Radioelektronika 2012, www.radioelektronika.cz.
11. R.C. Gonzales, R.E. Woods, Digital Image Processing, pp. 525-626, Pearson Prentice Hall, Upper Saddle River, New Jersey, 2008.
12. D. A. Huffman, "A method for the construction of minimum Redundancy codes," in Proc. IRE, Sep. 1952, Vol. 40, pp. 1098–1101.
13. G. Lakhani and V. Ayyagari, "Improved Huffman code tables for JPEG's encoder," IEEE Trans. Circuits Syst. Video Technol., Vol. 5, No. 6, pp. 562–564, Dec. 1995.
14. Adina Arthur, V. Saravanan, "Efficient Medical Image Compression Technique for Telemedicine Considering Online and Offline Application", International Conference on Computing, Communication, and Applications, 2012.
15. Yu-Ting Pai, Fan-Chieh Cheng, Shu-Ping Lu, and Shanq-Jang Ruan, "Sub-Trees Modification of Huffman Coding for Stuffing Bits Reduction and Efficient NRZI Data Transmission", IEEE Transactions on Broadcasting, vol. 58, no. 2, June 2012.
16. S.B. Choi and M.H. Lee, "High speed pattern matching for a fast Huffman decoder," IEEE Trans. Consum. Electron., vol. 41, no. 1, pp. 97–103, Feb. 1995.
17. Huang-Chih Kuo and Youn-Long Lin,"A Hybrid Algorithm for Effective Lossless Compression of Video Display Frames", IEEE Transactions on Multimedia, vol.14, no.13, June 2012.

AUTHORS PROFILE

C. Saravanan received Ph.D. degree in Analysis and Modelling of Gray-Scale Image Compression from National Institute of Technology, Tiruchirappalli, Tamilnadu, India in 2009. His research interest includes image compression, colour image processing, medical image processing, fingerprint analysis, and any image processing related areas. He has published 20 research articles in international and national journals, conferences, chapters in books. He started his career with the Bharathidasan University, Tiruchirappalli, Tamilnadu, India, in 1996. He continued his career with the National Institute of Technology, Tiruchirappalli, Tamilnadu, India. At present he is working as an Assistant Professor (System Manager) and HOD of Computer Centre at National Institute of Technology, Durgapur, West Bengal, India**.** He is a life member of Computer Society of India and Indian Society for Technical Education, member of IEEE, member of acm, member of ICSTE, IAENG.

M. Surendar received his B.Tech degree in Information Technology from Jyothishmathi Institute of Technology & science JNTUH), Karimnagar, Andrapradesh in 2009. He is currently pursuing M.Tech degree in Information Security in NIT Durgapur, West Bengal and working on images. He has published two papers in National conferences and his area of interest includes Digital Image processing, C Programming, Algorithms, and Compiler Design.