IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

nCORETech

LBS College of Engineering, Kasaragod

Vol. 3, Special Issue 1, February 2016

A Review on Lossless Compression Methods for Binary Images

Nithin P P[1], Sheeba K[2]

M. Tech Student, Dept. of ECE, LBS College of Engineering, Kasaragod, India [1]

Associate Professor, Dept. of ECE, LBS College of Engineering, Kasaragod, India[2]

Abstract: An efficient storage and transmission is a great challenge in digital imaging, since it requires large number of bits to represent an image. Image compression reduces the number of bits needed to represent an image. Lossless compression techniques are used in applications which cannot tolerate any difference between the original and compressed image. Binary image is used in many digital imaging applications such as document imaging, finger print databases and geographical information systems. This paper is a study of various lossless image compression techniques. A comparison of performance of different lossless compression algorithm is also made.

Keywords: Huffman Coding, Run length Coding, Arithmetic Coding, Lempel-Ziv-Welch Coding.

I. INTRODUCTION

A large number of bits are required to represent a digital image and image compression [1] is used to reduce the amount of memory required to store an image. The objective of image compression is to reduce redundancy of the data for an efficient storage and transmission. Image compression techniques are generally classified into lossless and lossy compression techniques. In lossless image compression technique [2], the image which is reconstructed after compression will be numerically identical to the original image. In lossless coding, every detail of the original data is restored upon decoding. It achieves modest amount of compression and reduces file size with no loss of quality. This type of compression can be applied to both images and video files where it reduces the size. It rewrites the data of the original file in an efficient way. When compared to lossy compression, the images and audio files which are compressed using lossless compression are larger because no quality is lost.

As in case of binary images, compression is concerned with reduction of bits that represents the image. Compression of binary image is vital due to enormous amount of images that are being processed and stored in a wide range of applications such as text, maps, fingerprint etc. Image compression has an important application in the field of image transmission and the image storage despite the large capacity storage devices that are currently available. Therefore to reduce the memory size and execution time, an efficient method for storing and transmitting an image is required. The general principle of image compression algorithms is transforming binary digits into a new one that specifies the same information but with fewer digits, so the file can be small as possible. The main benefits of image compression includes low storage space requirements, quicker sending and receiving images, and less time on image viewing and loading.

This paper is organised as follows. Section I gives an introduction to Lossless compression. Section II describes the various Lossless compression techniques. Section III provides the performance evaluation parameters. . Finally the conclusion is described section IV.

II. LOSSLESS IMAGE COMPRESSION TECHNIQUES

The lossless compression scheme [3], as shown in figure 1, the reconstructed image, after compression, is similar to the original image. However lossless compression can only achieve a modest amount of compression. Lossless compression technique can reduce the size to half. It depends on the file used for compression. This makes fast transmission of files across the networks because of reduced size.

In lossless image compression techniques images are considered as sequence of pixels in the row major order. Each pixel is processed with two separate operations. The first step consists of predicting the numerical value of the next pixel. A linear combination of neighboring pixel values are involved in a predictor in conjunction with an edge detection heuristic, that gives discontinuities in intensities. The second step consists of coding the difference between the actual pixel intensity of next pixel and the actual intensity.

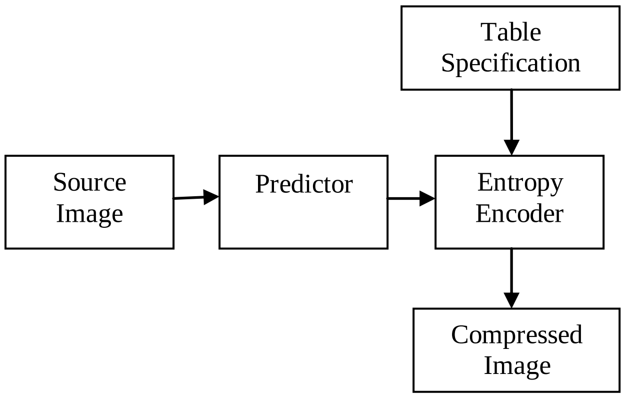

Fig. 1. Block Diagram for Lossless Compression

A. Huffman Coding

Lossless compression techniques were used in applications that cannot tolerate any loss of information. There are many techniques used for lossless compression of images. One of such best algorithm is Huffman coding. It is a variable length entropy coding technique. Huffman coding [4] is based on the idea that, symbols occurred with higher probability are represented with shorter length code word and symbols occurred with smaller probability are represented with longer code word. Thus the last two symbols with least probability of occurrence are represented with same code word length. They differ in one bit. This reduces the average code word length. Thus the code is optimum. It satisfies prefix property and hence it is uniquely decidable [5].

Code word are assigned to symbols by constructing the binary coding tree. Each symbol in the alphabet represents leafs of the binary coding tree. The code of a symbol in the alphabet is obtained by traversing the tree from root to leaf with zero or one is added depending whether the child of that node is left or right along the path. Each block in the codebook represents symbol of the alphabet with corresponding probabilities.

The Huffman algorithm can be explained as follows.
Step 1: Take the characters and corresponding frequencies, and sort this list by increasing frequency.
Step 2: All characters are the vertices of the tree
Step 3: Take the first 2 vertices from the list and make them children of a vertex having the sum of their frequencies.
Step 4: Insert the new vertex into the sorted list of vertices waiting to be put into the tree
Step 5: If there are at least 2 vertices in the list, go to step 3.
Step 6: Read the Huffman code from the tree

B. Run length Coding

RLE is one of the images compression techniques in which sequence of identical symbols are replaced with a pair containing the symbol and the number of repetitions of that symbol [6]. It is widely accepted compression technique in the fax standard. RLE [14] is mainly effective in compressing binary images since there are only two possible intensities (black and white). Additionally, a variable-length coding can be applied to the run lengths themselves. The approximate run-length entropy is

$$H_{RL}= H_0+ H_1 / L_0+ L_1$$

Where H0 and H1 are entropies of the black and white runs and L0 and L1 are the average values of black and white run lengths. Images with repeating intensities along their rows or columns can often be compressed by representing them as run-length pairs, where each run-length pair specifies the start of a new intensity and the number of consecutive pixels having that intensity. This technique is used for FAX data compression, in BMP file format, etc.
Input: AAABBCCCCDEEEEEEAAAAAAAA
Output: 3A2B4C1D6E8A

The first step is to read the files and then it scans for finding the repeating character strings. This method is useful for images having solid black pixels. This algorithm is also effective in case of long run of characters. But it is not much effective when data files have less repetition of characters.

C. Arithmetic Coding

The arithmetic coding is a lossless compression technique which generates non-block codes. So that a one to one correspondence between code words and source symbols does not occurs. But a single codeword can be used to

represent an entire sequence of symbols that corresponds to a real interval between zero and one. When the number of symbols in the message increases more number of bits is required for representing the interval. But the interval used to represent it will become smaller. Size of the intervals reduces with probability of occurrence each symbol. The main principle of this technique is providing an interval for each potential bit data [7]. The last symbol is first encoded and decoded in Arithmetic coding.
1. Arithmetic coding is a modern coding method than Huffman coding.
2. Huffman coding assign code word to each symbol which has an integral length of bits
3. Arithmetic coding takes whole string data as one unit.

Arithmetic coding is based on the following principles.

- The source alphabet should contain finite number of symbols
- All possible symbols should have finite length.
- The real numbers in the interval [0,1] can assign a unique subinterval for any input symbol sequence.

Arithmetic coding is often used technique for binary symbols. As each symbol (bit) begins the coding process, its context is formed in the context determination block. Arithmetic coding can be implemented in JPEG-2000, JBIG1, JBIG2, MPEG-4 AVC and H.264.

D. Lempel-Ziv Welch Coding
Lempel-Ziv-Welch (LZW) is a dictionary technique [8]. It maps a variable number of symbols into a fixed code. LZW places longer repeated entries in a dictionary. It emits the code for an element, rather than the string itself, if the element has already been placed in the dictionary. In dictionary technique a symbol or a string of an alphabet is represented with an index to the dictionary, created from source alphabet. If a new symbol is occurred, which is contained in the dictionary, and then it is coded with the index. Coding cannot be efficient if the new symbol was not contained in the dictionary. A dictionary initially contains single character string for all the possible characters. The algorithm is performed by scanning through the input strings, until it found a string that is not present in the dictionary. If a string is not contained in the dictionary, it will send the string without the last character, which is a member of the dictionary. The string is then stored as the new symbol in the dictionary. The avoided last string character is used for scanning of further substrings. These way subsequent longer strings are stored in the dictionary, and then made further processing of these strings as single output. Better compression performance is achieved in case of data containing repeated strings. Higher compression ratio is achieved for longer sequences.
The decoding algorithm performed by reading the encoded bits and outputting the corresponding strings from the dictionary. The next value is obtained at the same time, and added to the dictionary. The concatenation of strings forms the first character. The decoding continues by reading the next value in the input and the process is repeated until there are no more input values. Also the final value is decoded with no more additions into the dictionary. Hence the dictionary is built by the decoder, which is identical to the encoder and uses it for further decoding.

III.PERFORMANCE EVALUATION PARAMETERS

There are many ways to calculate the effectiveness of the compression. The most frequently used performance evaluation Parameters for this purpose is compression ratio (CR), compression time, mean square error(MSE) and peak signal to noise ratio(PSNR).

Compression Ratio: Compression Ratio is the ratio of uncompressed file size to the compressed file size [9].

$$CR = \frac{Uncompressed\ file\ size}{Compressed\ file\ size}$$

Compression time: Compression and decompression time is separately considered. The algorithm is time efficient when it takes less time for both compression and decompression.
Entropy: Entropy is measured as the average rate of self information associated with the source symbol. It is the average number of bits required to code the source outputs.

$$H = \sum_{i=1}^{n} p_i \times log_b \frac{1}{p_i}$$

Where p_i is the probability of occurrence of ith symbol.

Code efficiency: Code efficiency is measured in percentage and it is obtained by taking the ratio between the entropy and average length. Average number of bits required to represent a single code word is measured as the average length. It is given by

$$\acute{L} = \sum_{i=1}^{n} p_i \times l_i$$

where p_i is the probability of occurrence of ith symbol and l_i is the length of codeword of each symbol.
Code efficiency in percentage is given by

$$E = \frac{H}{\acute{L}}$$

Where H is the entropy of the source and Ĺ is the average code word length.

Table I Summarizing the advantages and disadvantages of various data compression algorithm

ENCODING METHOD	ADVANTAGES	DISADVANTAGES
Huffman Coding	Simple for coding characters [10] It gives smaller bits for more frequently appearing symbols [11]	Coding tree will not be recreated.
Run Length Coding	This algorithm is easy to implement [12]	RLE compression is only efficient with files that contain lots of repetitive data [12] White or black areas are more suitable than colored image [10]
Arithmetic Coding	Its ability to keep the coding and the modeler separate [13] No code tree needs to be transmitted to the receiver [13] It uses fractional values [13]	Arithmetic coding have complex operations because its consists of additions, subtractions, multiplications, and divisions [13] Arithmetic coding significantly slower than Huffman coding, there are no infinite precision [13] Two issues structures to store the numbers and the constant division of interval may result in code overlap [13]
LZW	Simple, fast and good Compression [10]	Actual compression hard to predict[10] LZW algorithm works only when the input data is sufficiently large and there is sufficient redundancy in the data

IV. CONCLUSION

Reviewed and summarized various techniques of lossless image compression for binary images and their performance evaluation measures are studied. To conclude, all the lossless image compression techniques of binary images are useful in their related areas. New compression techniques are developing, which gives better compression performance. This review paper gives clear idea about lossless compression techniques and different quality metrics for image compression. Conclude that the efficiency of compression algorithm depends on the quality of image, amount of compression and speed of compression.

REFERENCES

[1] David Salomon, Data Compression, The Complete Reference, 2nd Edition Springer-Verlag 1998.
[2] Ming Yang and Nikolaos Bourbakis, " An Overview of Lossless Digital Image Compression Techniques," IEEE,pp.1099-1102, 2005.
[3] Ming Yang, and Nikolaos Bourbakis, "An Overview of Lossless Digital Image Compression Techniques," IEEE Conference on Circuits and Systems, 2005
[4] Rupinder Singh Brar, and Bikramjeet singh, "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data," International Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE),Volume 3, Issue 3, March 2013.
[5] http://www.cprogramming.com/tutorial/computersciencetheory/huffman.html
[6] Majid Rabbani, Paul and W.Jones; "Digital Image Compression Techniques". Edition-4, page 51,1991
[7] A.S. Ragab, Abdalla S.A. Mohmed, and M.S. Hamid, "Efficiency of Analytical Transforms for Image Compression" 15th National Radio Science Conference, 1998.
[8] Dalvir Kaur, and Kamaljeet Kaur "Data Compression on Columnar-Database Using Hybrid Approach (Huffman and Lempel-ZivWelch Algorithm)," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013.
[9] Js.r. kodituwakku, u. and S.amarasinghe, "Comparison of Lossless Data Compression Algorithms for Text DATA", Indian Journal of Computer Science and Engineering, Vol 1 No 4 416-425.
[10] Jindal V. ,Verma A. K and Bawa S. "Impact of Compression Algorithms on Data Transmission ".
[11] "http://cs.gettysburg.edu/~skim/cs216/lectures/huffman.pdf""
[12] "http://www.prepressure.com/library/compressionalgorithm/rle"
[13] Iombo C." Predictive data compression using adaptive arithmeticcoding" AThesis (100117/unrestricted/Iombo_thesis.pdf
[14] H. Rosdolsky, T. Huang, and H. Mayer, "Optimum run length codes," IEEE Transactions on Communications, vol.22, pp.826-835,2003