RESEARCH ARTICLE OPEN ACCESS

Comparision Of Various Lossless Image Compression Techniques

Eshan Mishra[1], K.K. Singh[2]
Department of Electronics & Communication, Amity School of Engineering and Technology
Amity University Lucknow Campus

ABSTRACT
Today images are considered as the major information tanks in the world. They can convey a lot more information to the receptor then a few pages of written information. Due to this very reason image processing has become a field of research today. The processing are basically are of two types; lossy and lossless. Since the information is power, so having it complete and discrete is of great importance today. Hence in such cases lossless techniques are the best options. This paper deals with the comparison of different lossless image compression techniques available today.
Keywords: Lossless Compression techniques, Huffman coding, LZW coding

I. INTRODUCTION

An image is a visual perception of a subject or the surrounding, by a living being. But in the digital world, it is an organization of minute building blocks, called "*pixels*". These pixels are arranged in a two-dimensional array. Today, these digital images have become an essential element of the information grid. Hence, the transmission and storage of these information formats has gained a great attention form the digital-world worldwide.
An image comprises of two basic components, information or *data* and *redundant data*. . Of these components Data is the part of our concern and contains the actual useful information, whereas the redundant data is the unwanted information attached to the actual data. Data compression is a process that deals with these redundant data and works for its reduction or banishment.

1.1. Principle behind image compression

The pixels of a digital image are in harmony with one-another. This harmony within the pixels of a digital image generates redundant bits in the information packet. As a result various compression techniques are employed to deal with these bits. The property, induced in an image due to the redundant bits is called as "*Redundancy*".[1]
Redundancies are basically of three types:
(i) *Spatial redundancy*, it is due to the correlation between the neighbouring pixel values.
(ii) *Spectral redundancy*, it arises due to the correlation between different spectral bands.
(iii) *Temporal redundancy*, it exists where moving frames of images have a correlation among them.

II. COMPRESSION TECHNIQUES

The compression techniques basically emphasis on reducing the memory required to store and transmit an information packet. In order to do so they basically try to reduce or eliminate the redundant bits,but in doingso they some of the times destroy a part of the actual data, making it impossible to retrieve the information completely. Hence the compression techniques are broadly divided into two categories;

1. Loss-less compression technique and
2. Lossy compression technique.

3.1. Lossy compression technique:-

As the name indicates, this technique involves certain loss of the data, leading to discrepancy between the original data and the reconstructed data. But the compression ratio of the lossy technique is higher than that of lossless technique.

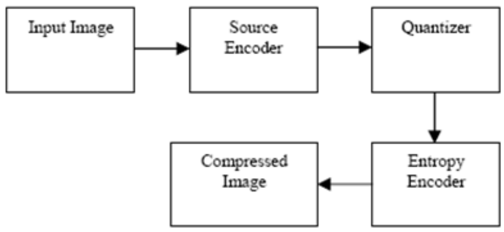

Figure 2.1(a). Block diagram for a lossy compression technique [2].

Some of the common examples of lossy compression techniques are; Discrete Cosine Transform, and Discrete Wavelet Transform.

3.2. Loss-less compression technique:-

As the name indicates, it's a compression technique which leads to no data loss. Hence the original data is identical to the reconstructed data. These techniques do not add any kind of noise to the data, hence also known as noiseless techniques. These techniques sometime use the tactics of statistic/decomposition to maintain a lossless result, hence are also known as the entropy coding [3]. The technique is used in many applications such as ZIP file format & in UNIX tool gzip.

Most of the lossless compression techniques comprises of two processes carried out in a sequence. These processes are; first generations of a *statistical model* for the input data, and the second step being the uses of this model to map input data to bit sequences in such a way that "probable" data will produce shorter output than "improbable" data[2].

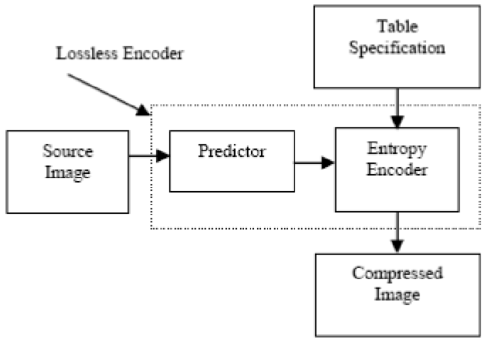

Figure 2.2 (a) Block of lossless Compression technique [2]

Some of the common examples of lossless compression techniques are; Huffman coding, LZW coding.

IV. HUFFMAN CODING

DR. David A. Huffman in 1952 introduced this data compression technique. “A method for the construction of minimum redundancy code.” Huffman coding is a formof statistical coding and it tries to minimize the number of bits required to represent a data. This compression technique permits symbols to vary in length. Code word lengths are no longer fixed like ASCII code. The word lengths vary and will be shorter for the more frequently used characters [4].

To visualize any particular encoding it is in best interest to picture it as a binary tree. The data is stored at a leaf node. Any particular data encoding is obtained by tracing the path from the root to its node. Each left-going edge represents a 0, each right-going edge a 1.

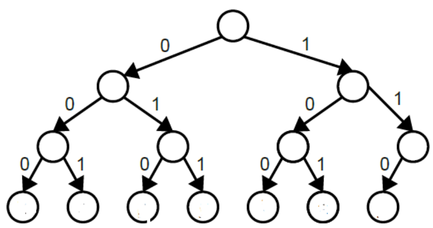

Figure 3(a) Binary Tree [5]

An algorithm for generating the optimal tree giving a minimal per-character encoding for a particular file was introduced by D. Huffman in 1952.

The general approach is as follows[5]:

1. Create a collection of singleton trees, with weight equal to the data frequency.
2. From the collection, pick out the two small weighted trees and eliminate them. Now combine them into a new tree whose root has a weight equal to the sum of the weights of the two trees and with the two trees as its left and right sub-trees.
3. Add the new combined tree back into the collection.
4. Repeat steps 2 and 3 until there is only one tree left.
5. The remaining node is the root of the optimal encoding tree.

The Huffman encoding also includes certain flaws that the encoder must know the probabilities of the data in the compressed files as this can require more bits to encode the file if this information is unavailable [4].

V. LWZ CODING

This lossless algorithm for compression was a team work of three grate scientists, Abraham Lempel, Jakob Ziv and Terry Welch, and hence the name Lempel-Ziv-Welch compression technique. The original Lempel-Ziv approach to data compression was first published in 1977. Later Welch’s refinements were published in 1984[6].This compression algorithm is simple, lossless and dictionary based. Dictionary based algorithms scan a file and search the sequences of data or string that occur more than once in that file. LZW compression technique replaces strings of characters with single codes without analysing the incoming data. It adds every new found characters of string in the dictionary and data compression occurs on the single code [7].

The encoding and decoding process for the LZW algorithm is as given below [7];

LZW Encoding:

1. Initial table with initial character strings
2. P=first input character
3. WHILE not end of input stream
4. C=next input character
5. IF P+C is in the string table
6. P=P+C
7. ELSE
8. output the code for P
9. add P+C to the string table
10. P=C
11. END WHILE
12. output code for P

LZW Decoding:

1. Initialize table with single character strings
2. OLD = first input code
3. output translation of OLD
4. WHILE not end of input stream
5. NEW = next input code
6. IF NEW is not in the string table
7. S = translation of OLD
8. S = S+C
9. ELSE
10. S = translation of NEW
11. output S
12. C = first character of S
13. OLD + C to the string table
14. OLD = NEW
15. END WHILE

LZW compression is the best technique for reducing the size of files containing more repetitive data.

VI. COMPARISON BETWEEN HUFFMAN AND LZW CODING

S.NO.	PROPERTY	HUFFMAN CODING	LZW CODING
1.	**Working principle**	distinct symbols have distinct probabilities of incidence	many groupings of pixels are common in images
2.	**Compression ratio**	70-75%	83-87%
3.	**Speed(Encoder/Decoder)**	990/1300	5700/8400
4.	**Memory(Encoder/Decoder)**	48/12	8/8
5.	**Hardware Decoder Components**	4	2
6.	**Parallel Decompression**	No	Yes
7.	**Decompression Bandwidth**	1 byte per iteration	Average 1.36-1.72 bytes per iteration
8.	**Tables involved**	Look-up tables	String tables

Table 5. Comparison between Huffman and LZW Coding[8],[9],[10],[11]

VII. CONCLUSION

On the basis of the above comparisons, I can conclude that if one is considering speed, compression ratio, memory space needed and compactness of the hardware than LZW coding leads the way. It also favours parallel decompression, which is another set-back for Huffman coding. The only area whereHuffman coding is ahead of LZW coding is that of decompression bandwidth, where too it leads with a marginal difference.Some of the other differences between them include their working principle and the type of the table involved in there processing. As for the conclusion, it can be said that LZW Coding Technique is quite superior to Huffman Coding Technique.

REFRENCE:

[1] Bhawan Gautam, "Image compression using discrete cosine transform & discrete wavelet transform", Department of Computer Science and Engineering, National Institute of Technology, Rourkela, May 2010
[2] Manjinder Kaur, Gaganpreet Kaur, "A Survey of Lossless and Lossy Image Compression Techniques", *IJARCSSE,* International Journal of Advanced Research in Computer Science and

Software Engineering,Volume 3, Issue 2, February 2013 ISSN: 2277 128X.
[3] Sonal, Dinesh Kumar, "A Study Of Various Image Compression Techniques", Department of Computer Science & Engineering, Guru Jhambheswar University of Science and Technology, Hisar.
[4] Mridul Kumar Mathur, Seema Loonker, Dr. Dheeraj Saxena, " Lossless Huffman coding Technique for Image Compression and Reconstruction using Binary Trees", IJCTA | JAN-FEB 2012, ISSN:2229-6093.
[5] Julie Zelenski, Keith Schwarz, "Huffman Encoding and Data Compression", CS106B Spring 2012, May 23, 2012.
[6] N Santosh Arjun, Atul Negi, "An Approach to Self-embedding Document Watermarking Based on LZW Algorithm", IEEE 1-4244-1272-2/07, February 2007.
[7] Dalvir Kaur, Kamaljit Kaur, "Huffman Based LZW Lossless Image Compression Using Retinex Algorithm", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013.
[8] Timothy c. Bell, "Better OPM/L Text Compression", IEEE TRANSACTIONS ON COMMUNICATIONS, vol. Com-34, no. 12, December 1986.
[9] Talal Bonny and Jörg Henkel, "FBT: Filled Buffer Technique to Reduce Code Size for VLIW Processors", IEEE 978-1-4244-2820-5/08, May 2008.
[10] Marwan Y. Al-Saiegh and Sridhar Krishnan," Fixed Block-based Lossless Compression of Digit a1 Mammograms", IEEE Vol. 2, DOI: 10.1109/CCECE.2001.933567, page(s) 937-942, 2001.
[11] Chang Hong Lin, Yuan Xie, Wayne Wolf, " LZW-Based Code Compression for VLIW Embedded Systems", IEEE, Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum (DATE'04), 1530-1591/04, April 2004.