International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 2 Issue: 12

ISSN: 2321-8169
3896 – 3900

Image Compression Using Lossless Compression Techniques

Gaurav Gupta
Department of computer science and engineering
Shoolini University HP
Solan, India
gaurav@shooliniuniversity.com

Parul Thakur
Department of computer science and engineering
Shoolini University HP
Solan, India
parul.thakur51@gmail.com

Abstract**—** Image compression is an application of data compression that encodes the original image with few bits. Data compression method reduces the size of data by reducing irrelevancy and redundancy of the image data, so data can store and transmit in an efficient form. The rapid growth in the field of multimedia and in digital imaging need to transmit digital images, transmitting the images without compressing them takes more disk space as well as time for transmission over the network. In Lossless image compression no data loss when the compression technique is done. In this paper we discuss the basic introduction about image compression, various types of lossless image compression techniques. Some of the lossless techniques are: Huffman coding, Run Length encoding arithmetic coding and LZW are discussed and lastly the performance parameters and benefits of the image compression.

Keywords- Compression, Lossless, Lossy, Huffman, Run-length encoding, LZW,Area Coding.

*****

I. INTRODUCTION

An image is in the form of matrix of square pixel arranged in columns and rows. There are different types of images, Binary image, RGB image, Intensity image. Compression is the method used to reduce the size of data or it is technique to decrease the number of bits required to represent text, image or video sequence or music. It is a core module for storing and sending data; hence it has a gamut of application.[4] Image compression is an application of data compression which reduces the size of data by reducing irrelevancy and redundancy of the image data, so data can store and transmit in an efficient form.[3] Digital image is basically two dimensional array of pixels. Digital image require more space for storage and also required large bandwidths for transferring the data. Applications where image compression play important role like medical imaging, facsimile transmission (FAX), remote sensing and document which depend on the efficient storage and transmission of binary, gray scale or RGB Images, colour images[9]. The primary goal of image compression is reduces the transmission time and save the storage space so it can also decreases the cost of storage.[2] Reducing the memory footprint of image data will minimize bandwidth consumption ,Mostly web documents contain image data so it is important that the when image data will be transferred over the network it will take reasonable time frame. The rapid growth in the field of multimedia and in digital imaging need to transmit digital images, transmitting the images without compressing them takes more disk space as well as time for transmission over the network. In Lossless image compression no data loss when the compression technique is done.

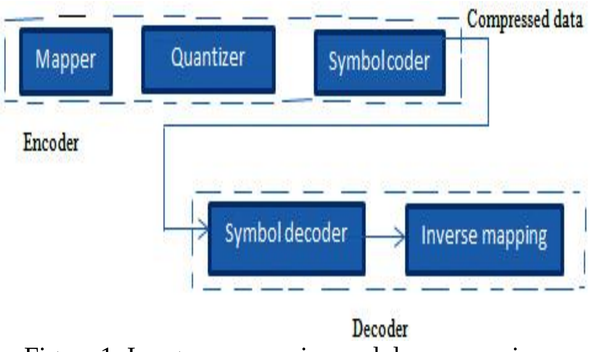


Figure 1. Image compression and decompression

1. *Principle of Compression*

A common characteristic of most of the images is that the neighbouring pixels are correlated and therefore contain redundant information. The foremost task is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction [15]. Redundancy reduction aims at removing duplication from the signal source (image). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancy can be identified:

1. *Spatial Redundancy* or correlation between neighbouring pixel values.
2. *Spectral Redundancy* or correlation between different color planes or spectral bands.
3. *Temporal Redundancy* or correlation between adjacent frames in a sequence of images (in video applications).

Since, we focus only on still images. Image compression techniques are explored in this paper. For image compression there are three types of redundancies,
1. Coding Redundancy
2. Interpixel Redundancy
3. Psychovisual Redundancy
Coding redundancy is present when less than optimal code words are used. Interpixel redundancy results from correlations between the pixels of an image

II. PERFORMANCE PARAMETERS

There are two performance parameters which are used to measure the performance of the image compression algorithm. These are following:

1. Peak signal to noise ratio (PSNR)
2. Mean square error (MSE).

PSNR is the measurement of the peak error between the compressed Image and original image. The higher the PSNR contains better quality of image.
To compute the PSNR first of all mean square error (MSE) is computed. Mean Square Error (MSE) is the cumulative difference between the compressed image and original image. Small amount of MSE reduce the error and improves image quality.

$$MSE = \sum_{MN} [I_1(m,n) - I_2(m,n)^2 / M * N$$

Here M and N are the number of rows and columns in the input images.
The PSNR is computed from following equation:

$$PSNR = 10 log_{10} \left(\frac{R^2}{MSE}\right)$$

III. IMAGE COMPRESSION TECHNIQUES

The image compression techniques can be broadly classified into two categories as Lossless and Lossy Compression. In Lossless Compression the original image can be reconstructed from the compressed image. In lossless compression technique, as name indicate that no loss of information. When we need certainty that we achieve the same what we compressed after decompression, lossless compression methods are the only choice. This technique is used for application that cannot tolerate any difference between the original and reconstructed data. Lossless compression is used for storage related application these applications **include**: file zipping, compression of audio, image/videos, source code, text etc.
Lossless Compression Techniques

1. Huffman Coding
2. LZW coding
3. Run Length Encoding
4. Area Coding
5. Arithmetic Coding

In lossy compression technique as name indicates that some loss of information this technique is basically used for network related application. The data that have been compressed using lossy techniques generally cannot be recovered or reconstructed exactly.[10]

Lossy Compression Techniques

1. Fractal Coding
2. Block Truncation Coding
3. Sub band Coding
4. Transformation Coding
5. Wavelet Scalar Quantization
6. S3 Textual Compression.

Here we have discussed various lossless compression techniques, the latter consist of three steps [1];

1. Transformation: convert the image data into a form that can be compressed more efficiently by further stages.
2. Data to symbol mapping: the input data into symbol that can be efficiently coded by last stage
3. Lossless symbol coding: generate a binary bit stream by assigning binary code words to the input symbol.

1(a). Huffman Coding Technique.

Huffman coding technique is popular technique for creating prefix-free codes. It is the technique for removing coding redundancy. It is a fixed-to-variable length code, that is, it maps fixed length input symbols to variable length code words. Huffman coding was proposed by Dr.David A. Huffman in 1952.Using this procedure codes generated are called Huffman codes. Huffman coding is a statistical coding which reduces the number of bits required to represent a string of symbols. This algorithm allows symbols to vary in length. In this technique symbols having higher probability of occurrence will have shorter code word and symbols having less probability of occurrence will have large code word.[11]

Huffman Technique depends upon two observations regarding optimum prefix codes.

1. In this technique symbols having higher probability of occurrence will have shorter code word and symbols having less probability of occurrence will have large code word.
2. In an optimum code, the two symbols that occur least frequently will have the same frequency.

We consider here only codes in which no code word is also a prefix of some other code word. Such codes are called ***prefix codes***. Encoding is very simple for binary character code so we focus on code words representing each character of the file. In below example we code the 4-character file abcd as 0.101.100.111=0101100111,where (.) denotes concatenation.

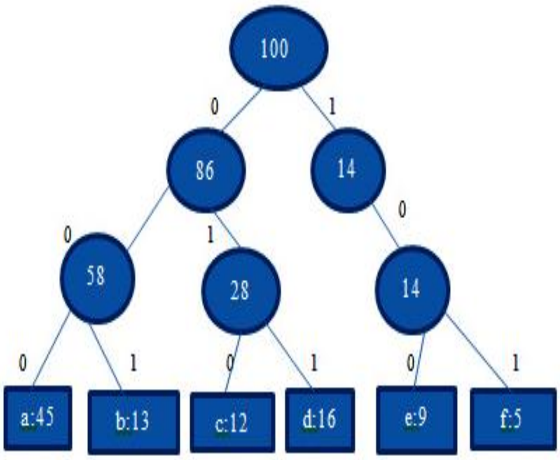
 ISSN: 2321-8169
3896 – 3900

	a	b	c	d	E	F
Frequency	45	13	12	16	9	5
Fixed-length code word	000	001	010	011	100	101
Variable-length code word	0	101	100	111	1101	1100

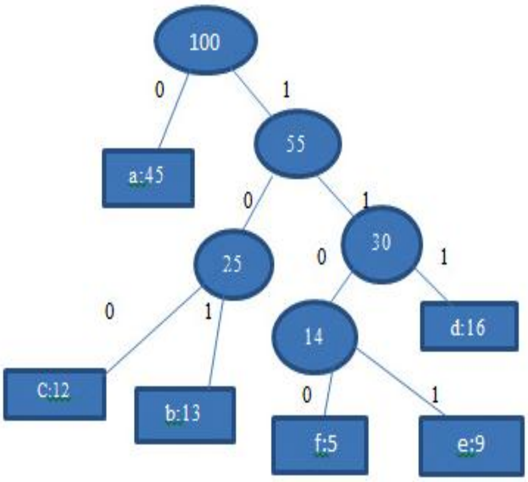
Table 1 Different character with different frequency of occurrence.

(b).Huffman Decoding

After the code has been created, coding / decoding is accomplished in a simple look-up table manner. The code generated is also called decodable block code. Where each input 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. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. After decoding the above Huffman code word by using the Huffman tree we get the original message.



(a) Tree corresponding to the fixed-length code.



(b) The tree corresponding to the optimal code

Characteristics of Huffman Coding:

1. In Huffman coding we can transfer large amount of information in lesser time and also store in small storage space.
2. Huffman codes are Prefix tree binary code trees.

2. Lempel Ziv Welch (LZW)

Lempel, Ziv and Welch (LZW) compression name is originate from the scientists Abraham Lempel, Jakob Ziv and Terry Welch. LZW compression algorithm is Simple, lossless and dictionary based compression algorithm. Lempel Ziv Welch (LZW) is “Dictionary based” compression method. In this method variable number symbols are maps to a fixed length code. LZW algorithm based on dictionary based compression technique scan a file for long sequence of data that occur more than once than LZW places longer and longer repeated sequence of data into dictionary. LZW compression algorithm replaces sequence of data in a single code. The code generated by LZW can be of random length. LZW compression works by replacing strings of characters with single code without doing any analysis of the incoming text data. It adds every new found characters of string in the dictionary and data compression occurs on the single code. Like any adaptive/dynamic compression method, this method is also start with an initial model, then in second step read each part of data, and in the last step update the model and encode the data . LZW is a "dictionary"-based compression algorithm; this means that as in Huffman coding characters count occur in table manner and building tree ,it does not occur in LZW algorithm. In this method data is encoded by referencing dictionary. Hereby when we want to encode a substring, in the output we write only a single code, that code is occur in the dictionary corresponding to substring’s index. This algorithm is best for text files. LZW algorithm is widely used in computer industry and in UNIX compress command is used for compression [13]

These strings are then stored in a dictionary and the compressed file with references are put wherever repetitive data occurred. The replaced code must contain more bits in it as compare to single character. The first 256 codes when using eight bit characters are initially assigned to the standard character set and the remaining codes are assigned to strings as the algorithm proceeds. LZW is an adaptive technique. As the compression algorithm runs, a changing dictionary of the strings that have appeared in the text so far is maintained. Because the dictionary is pre- loaded with the 256 different codes that may appear in a byte, it is guaranteed that the entire input source may be converted into a series of dictionary indexes. [3]

```
N: = NIL;
While (there is input){
M: = next symbol from input;
If (NM exists in the dictionary) {
N: = NM ;} else {
Output (index (w));
Add NM to the dictionary;
N: = M}}
```

Dictionary Creation in LZW Algorithm [7].

N is the dictionary of string starting at the current position the inner loop finds this match. The iteration then
Outputs the index for w and adds the string NM to the dictionary, where M is the next character after the match.

3. Run Length Encoding

Run length coding is a popular data compression technique. In this method longest sequence of the same symbol is replace by shorter sequence. Run length Encoding is a lossless compression technique of input data based on sequence of identical values. This technique is most useful on data that contains such things: simple graphic images such as icons, line drawings, and animations and is not useful for files that don't have any things. Run length encoding is not for any particular type of data, but is suited for compressing any type of data information, but in run-length encoding compression ratio depends upon the content of the data. This method is easy to implement and execute but most RLE algorithms cannot provide high compression ratio.
There are a number of variant of run-length encoding. Run length encoding mapps the image data as 1-Dimensional stream rather than as a 2-Dimenstional.and this method encode the image data in sequential process.[10]

4. Arithmetic Coding

In this technique, instead of coding each symbol separately, whole image sequence is coded with a single code. Thus, the correlation on neighboring pixels is exploited. Arithmetic coding is based on the following principles:

1. The symbol alphabet is finite.
2. All possible symbols sequences of a given length are finite;
3. All possible sequences are count ably infinite;
4. The number of real numbers in the interval [0,1] is unaccountably infinite; we can assign a unique subinterval for any given input (sequence of symbols)[8].

Unlike the Huffman coding, arithmetic coding does not provide one-to-one correspondence between source symbol and code words. In arithmetic coding, an entire sequence of source symbols is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between '0' and '1'.

The basic advantages of arithmetic coding are:

1. When applied to independent and identical distributed sources, the compression of each symbol is probably optimal.
2. It is effective in a wide range of situations and compression ratios.
3. It simplifies automatic modelling of complex sources, yielding near-optimal 'or' significantly improved compression for sources that are not independent and identically distributed.

But there are some disadvantages also due to which arithmetic codes are getting less popularity such as.

1. Sometime, they are very complex for coding and decoding.
2. Patents covered the most efficient implementation. Royalties and fear of patent infringement discouraged arithmetic coding in commercial products.
3. Efficient implementation was difficult

5. Area Coding

Area encoding is an enhanced method of run length coding. It mainly focuses on two dimensional characteristics of an image. This is one of the finest techniques out of all lossless techniques discussed before.[5] In this method the whole image is divided into blocks of size m*n pixels, which are classified as block having only white pixels, block having only black pixels or block with mixed intensity. This approach takes advantage of the anticipated structural patterns of the image to be compressed. As few solid black areas are expected, they are grouped with the mixed regions, allowing a 1-bit code word to be used for the highly probable white blocks. Area coding is highly effective and it can give better compression ratio but it has certain limitation that it can be applied to nonlinear transformation.[14]

BENEFITS OF COMPRESSION

1. Storage Space
The biggest benefit of compression is storage as when we compress the data it increases the storage space by reducing the size of file. It allows storage of more files in the available space. Lossless compression compress file up to 50% of its input file. Zip file technology uses lossless compression technique.[11]

2. Bandwidth and Transfer Speed
The benefit of compression is increases transfer speed. When we download a file such as text file, ppts or audio/video file, thus this download process uses network bandwidth. Bandwidth is the speed at which the network transfers the data. So compressed files take less bandwidth when we send or download the file as compare to uncompressed file and it can also take less time to send or download the file. Thus we can say that compression increases the transfer speed. Suppose we have a file of size 10 megabits and we have 1Megabit/ sec available bandwidth we want to download that file it will take 10 sec, But if file is compressed it will take 5 sec for downloading that file.[11]

3. Cost
Cost is also the benefit occur by using compression. Compression reduces the storage space so it also reduces the cost of storage. In the available storage space we can store more compressed data. We will not need to buy the extra storage sources like hard drive if we compress the data files to 50% of the input file [11].

4. Accuracy
Compression technique reduces the size of data so in compressed file fewer bits are transferred. It also decreases the transmission error.

ISSN: 2321-8169
3896 – 3900

5. Security

Compression also provides a level of security against Illegal Inquiry.

V. CONCLUSION

For data compression lossless and lossy techniques can be implemented. This paper gives clear idea about image compression and lossless compression techniques. All the image compression techniques are useful in their related areas and every day new compression technique is developing which gives better compression ratio. The compression algorithm depends on the three factors: quality of image, amount of compression and speed of compression. Hence we conclude that in lossy compression there is some loss of data, but in lossless compression there is no loss of data so if we use lossless compression we get the same output, without any loss of information and the techniques explained in this paper are important for image compression.

REFERENCES

[1] A Conrad Bovik.'The essential guide to image processing', vol 16,pp.385- 419,2009.
[2] D .Salomon,"Data Compression, the Complete Reference", 2nd Edition Springer-Verlag 1998.
[3] D. Kaur, K. Kaur, & Sahib, F. "Huffman Based LZW Lossless Image Compression Using Retinex Algorithm".
[4] Gonzalez and Woods, "Digital image processing", 3rded.,Vol 8,pp.409-518,2007.
[5] I. M.Shilpa,and D.Margate."A Review on-Lossless Image Compression Techniques and Algorithms."
[6] M.Sharma "Compression Using Huffman Coding" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.
[7] M. Mridul Kumar, S. Loonker,and D.Saxena." Lossless Huffman coding technique for image compression and compression and reconstruction using binary trees." International Journal of Computer Technology & Applications3.1 (2012).
[8] P..Thiruvallur, and T. N. Thiruvallur." Images and Its Compression Techniques–A Review." Int. J. of Recent Trends in Engineering and Technology2.4 (2009).
[9] R C. Gonzalez, Richard E Woods ―"Digital image processing"‖, Second Edition page 410.
[10] S. Palvee, and R. Mahajan. "A review on compression technique with run-length encoding." International Journal (2013).
[11] www.ehow.com › Internet.
[12] www.TheLZWcompressionalgorithm.html
[13] www.cs.duke.edu/csed/curious/compression/lzw.html)
[14] www.cis.cs.technion.ac.il/Done_Projects/Projects_done/VisionClasses/DIP_1998/Lossless_Compression/nod.
[15] S.Saha; "Image Compression – from DCT to Wavelets: A Review Unpublished www/acm/org/crossroads/xrds6-3/sahaimgcoding.html.