Lossless BMP Image Compression with New Efficient Arithmetic Approaches

SuranjanGanguly
Department of Computer Science and Engineering
Jadavpur University
Jadavpur, Kolkata-700032, West Bengal, India
suranjanganguly@gmail.com

AnupamBera
Department of Information Technology
NetajiSubhash Engineering College
Kolkata 700152, West Bengal, India
bera.anupam@yahoo.in

AnirbanKundu
Kuang-Chi Institute of Advanced Technology
Shenzhen,Guangdong, P.R.China
anirban.kundu@kuang-chi.org, anik76in@gmail.com

Abstract

In this paper efficient approaches are made out of several approaches for developing a image compression [1][2] algorithm based on use of arithmetic approaches. Image processing technologies are used to pre-process the images which upon which the arithmetic rules are applied. Image data are stored and then these data are processed. These values are expressed in the power and remainder of 2. Then Huffman [8] and LZW (Lempel- Ziv – Welch) algorithms are used on these processed data. Here a satisfactory result is achieved. Then the extracted image value is dived these values by 26 and used the quotient and remainder portion of this arithmetic to compress image by using Huffman and RLE algorithms separately, on quotient part first RLE (Run Length Encoding) algorithm is applied then Huffman algorithm is used and on remainder part Huffman algorithm is applied. Another approach is used by spreading the extracted value from the image to ARGB to form a new compressed image. BMP (Windows bitmap) and JPEG (Joint Photographic Experts Group) file formats are considered for this compression [2][4] algorithm. The maximum success rate of compression of this new arithmetic based approach is 27.75 % on BMP images.

Keywords: image compression, LZW algorithm, Huffman algorithm, RLE algorithm, ARGB, JPEG, BMP

1. Introduction

Lossless Image compression [13] is required in application where images are subject to further processing and without changing the original data. We are mainly dealing with the compression technique after processing image formats (JPEG/JPG and BMP). We need image compression [7] for the following reasons:

Image compression is important for web designers who want to create faster loading web pages which make the websites more accessible to others. Image compression also saves lot of unnecessary bandwidth by providing high-quality image with fraction of file size.

Image compression is also important for people who attach photos to emails which will send the email more quickly, save bandwidth costs.

For digital camera users and people who save lots of photos on their hard drive, Image Compression is more important. By compressing image, we can store more images on our hard disk thus saving the memory space.

Though there are different algorithms which already exist for image compression, it is tried to search more improved version above all these algorithms i.e. it will compress an image in lossless mode, which is already compressed (JPEG/JPG is already compressed form and we want to implement a compression version which will work on JPEG/JPG) and BMP images. In the world of digital photography the major used file formats are only JPEG and BMP.

1.1 Image

An image [1] is essentially a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form. A digital image is basically a 2-Dimensional array of pixels.

1.2 Image Compression

Image compression [1] addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements. Compression is achieved by the removal of one or more of the three basic data redundancies:

1.2.1 Coding Redundancy

1.2.2 Interpixel Redundancy

1.2.3 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.

Psychovisual redundancy is due to data that is ignored by the human visual system (i.e. visually non-essential information).prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

1.3 Image Compression Considerations

Computer graphics applications, particularly those generating digital photographs and other complex colour images, can generate very large file sizes. Issues of storage space and the requirement to rapidly transmit image data across networks and over the Internet, have therefore led to the development of a range of image compression techniques, to reduce the physical size of files.

Most compression techniques are independent of specific file formats – indeed, many formats support a number of different compression types. They are an essential part of digital image creation, use, and storage.

However, a number of factors must be considered when using compression algorithms:

1.3.1 Efficiency:

Most algorithms are particularly suited to specific circumstances, which must be understood if they are to be used effectively. For example, some are most efficient at compressing monochrome images, whilst others yield best results with complex color images.

1.3.2 Lossiness:

Graphics compression algorithms fall into two categories:

Lossy compression [8] achieves its effect at the cost of a loss in image quality, by removing some image information.

Lossless compression [8] techniques reduce size whilst preserving all of the original image information, and therefore without degrading the quality of the image.

Although lossy techniques may be very useful for creating versions of images for day-to-day use or on the Internet, they should be avoided for archival master versions of images.

1.3.3 Openness:

Some compression algorithms are patented and may only be used under license. Others have been developed as open standards. This can be an important consideration in terms both of creation costs and long-term sustainability. The patenting of compression algorithms is a complex and controversial issue which is beyond the scope of this Guidance Note. However, known issues are highlighted.

1.4 Image Compression Techniques

The image compression techniques[1] are broadly classified into two categories depending whether or not an exact replica of the original image could be reconstructed using the compressed image.

These are:

1.4.1 Lossless technique

1.4.2 Lossy technique

Lossless compression technique

In lossless compression techniques[1], the original image can be perfectly recovered from the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image).It is also known as entropy coding since it uses statistics/decomposition techniques to eliminate/minimize redundancy

Lossless compression is used only for a few applications with stringent requirements such as medical imaging. Figure 1 states a straight approach of lossless image compression technique.

Fig. 1 Outline of lossless image compression

Following techniques are included in lossless compression:
1. Run length encoding
2. Huffman encoding
3. LZW coding
4. Area coding

Lossy compression technique

Lossy schemes provide much higher compression ratios than lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate for most applications .By this scheme, the decompressed image is not identical to the original image, but reasonably close to it.

Lossy compression techniques are consists of following steps: prediction – transformation – decomposition process is completely reversible .The quantization process results in loss of information. The entropy coding after the quantization step, however, is lossless. The decoding is a reverse process. Firstly, entropy decoding is applied to compressed data to get the quantized data. Secondly, dequantization is applied to it & finally the inverse transformation to get the reconstructed image. Major performance considerations of a lossy compression scheme include:

1. Compression ratio

2. Signal - to – noise ratio

3. Speed of encoding & decoding.

Lossy compression techniques includes following schemes:

1. Transformation coding
2. Vector quantization
3. Fractal coding
4. Block Truncation Coding
5. Sub-band coding

The table I summarises the lossiness of the algorithms described,and circumstances in which they are most efficient:

Table 1: Summary of algorithms

Algorithm	Lossiness	Efficient with
RLE	Lossless	Monochrome or images with large blocks of color
LZ Compressors	Lossless	All images
Huffman Encoding	Lossless	All images
LZW	Lossless	All images
JPEG	Lossy (lossless extension available)	Complex, True Colour images
JPEG 2000	Lossy, lossless supported	Complex, True Colour images

2. Problem Statement

2.1 BMP image file format

Image file formats [3] are standardized means of organizing and storing digital images. Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.

The BMP file format (Windows bitmap) handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large; the advantage is their simplicity and wide acceptance in Windows programs.

2.2 Huffman encoding algorithm

Developed by David Huffman in 1952, Huffman encoding [2][3][4][11] is one of the oldest and most established compression algorithms. It is lossless and is used to provide a final compression stage in a number of more modern compression schemes, such as JPEG and Deflate. In modified form, it is also used in CCITT Group 3 compression.

This is a general technique for coding symbols based on their statistical occurrence frequencies (probabilities). The pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the (binary) code of any symbol is not the prefix of the code of any other symbol. Most image coding standards use lossy techniques in the earlier stages of compression and use Huffman coding as the final step.

Huffman coding is a popular lossless Variable Length Coding (VLC) based on the following principles:

Shorter code words are assigned to more probable symbols and longer code words are assigned to less probable symbols.

No code word of a symbol is a prefix of another code word. This makes Huffman coding uniquely decodable.

Every source symbol must have a unique code word assigned to it.

We are now going to present a very important theorem by Shannon, which expresses the lower limit of the average code word length of a source in terms of its entropy. Stated formally, the theorem states that in any coding scheme, the average code word length of a source of symbols can at best be equal to the source entropy and can never be less than it. The above theorem assumes the coding to be lossless and the channel to be noiseless.

If m (z) is the minimum of the average code word length obtained out of different uniquely decipherable coding schemes, then as per Shannon’s theorem, we can state that

$$m(z) \geq H(z)$$

2.2.1 Measurement of source entropy [4]

If the probabilities of the source symbols are known, the source entropy can be measured using equation. Say, we have five symbols a1, a2, a3, a4, a5 having the following probabilities:

$P(a_1) = 0.2$, $P(a_2) = 0.1$, $P(a_3) = 0.05$, $P(a_4) = 0.6$, $P(a_5) = 0.05$

Using equation the source entropy is given by:

$H(z)$ = -0.2log0.2-0.1log0.1-o.o5log0.05-0.6log0.6-0.05log0.05 bits

=1.67 bits

2.2.2 Coding Efficiency

The coding efficiency [2] (η) of an encoding scheme is expressed as the ratio of the source entropy H(z) to the average code word length L(z) and is given by

$$\eta = \frac{H(z)}{L(z)}$$

Since $L(z) \geq H(z)$ according to Shannon's Coding theorem and both L(z) and H(z) are positive, i.e. $0 \leq \eta \leq 1$

It can be stated that Huffman coding is a popular lossless Variable Length Coding (VLC) based on the following principles:

Shorter code words are assigned to more probable symbols and longer code words are assigned to less probable symbols.

No code word of a symbol is a prefix of another code word. This makes Huffman coding uniquely decodable.

Every source symbol must have a unique code word assigned to it.

2.3 Lempel-Ziff-Welch (LZW) algorithm

Basic Principles of Lempel-Ziv Coding: We now consider yet another popular lossless compression scheme, which is originally called Lempel-Ziv coding, and also referred to as Lempel-Ziv-Welch (LZW) coding, following the modifications of Welch for image compression. This coding scheme has been adopted in a variety of imaging file formats, such as the graphic interchange format (GIF), tagged image file format (TIFF) and the portable document format (PDF). The basic principles of this encoding scheme are:

It assigns a fixed length codeword to a variable length of symbols.

Unlike Huffman coding and arithmetic coding, this coding scheme does not require a priori knowledge of the probabilities of the source symbols.

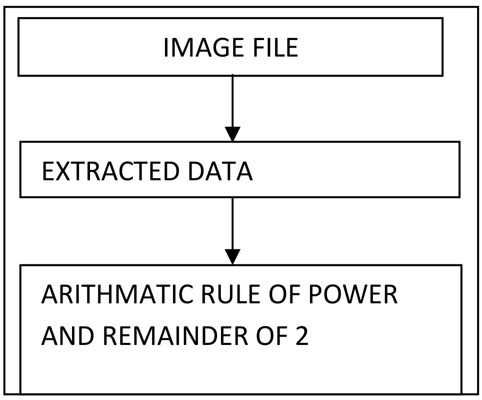
The coding is based on a "dictionary" or "codebook" containing the source symbols to be encoded. The coding starts with an initial dictionary, which is enlarged with the arrival of new symbol sequences.

There is no need to transmit the dictionary from the encoder to the decoder. A Lempel-Ziv decoder builds an identical dictionary during the decoding process.

3. Proposed System

There were many arithmetical approaches for the pre-process the image for compression purpose. Three different approaches from our research work have been discussed below:

3.1



After data of the image is extracted the arithmetic calculation for calculating the power of a number with base 2 is calculated with remainder. The image data is ranges from 0(zero) to 255 i.e. 256 distinct values are there. Now this 256 distinct values are reduces 128 distinct numbers ranges from 0(zero) to 128. After this huge distinct number reduction Huffman algorithm and LZW algorithm is applied to these data separately and result is discussed in table 2 of section 4.

3.2

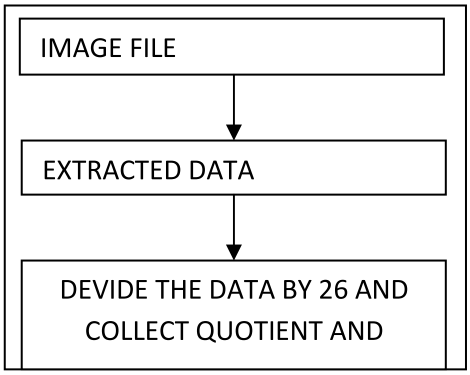


Fig. 3 Outline of proposed algorithm 2

On extracted image data these 256 distinct values are divided by 26. Remainder and quotient of this arithmetic approach is used for compression of image. In quotient part 0(zero) to 9 data i.e. 10 distinct data and remainder part consist of 26 distinct data which can be effectively mapped into 'A' to 'Z' according to the ASCCI (American Standard Code for Information Interchange) code and Huffman algorithm and LZW algorithm is applied to these data separately and result is discussed in table 2 of section 4.

3.3

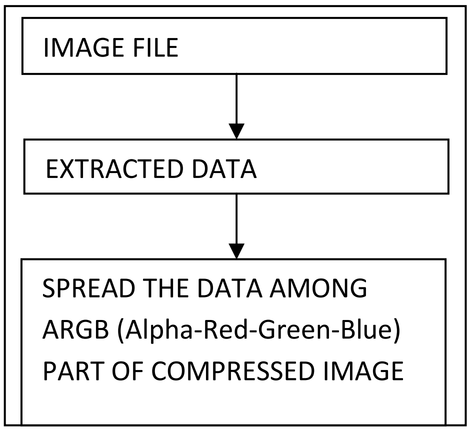


Fig. 4 Outline of proposed algorithm 3

An image is formed by number of pixels (picture element). It consists of 4 different elements like, Red-Green-Blue and Alpha i.e. transparency or they are called ARGB. Here the approach is based on that the extracted image data is distributed among ARGB part of new image. Four sequential data element of source image is combined into single value of new image to compress the source image to destination image and also to decompress back to original image from destination image to source image. The result of this arithmetic approach is discussed in table 2 of section 4.

4. Results

Figure 5 is a bmp file format image which is one of the experimental images for our research work and the arithmetic approaches shows that after successful decompression the original image comes back with 0% loss of information of the image. Following shows the comparative study of proposed system.

Fig. 5 Sample experimented image

Table 2: Summary Experimental Results

Proposed algorithm		Size before compression	Size after compression	Percentage (%)of compression
1	i)	26.3 kb	32.44 kb	-23.35 %
	ii)	26.3 kb	23.35 kb	11.21 %
2	i)	26.3 kb	21.43 kb	18.51 %
	ii)	26.3 kb	19 kb	*27.75 %*
3		26.3 kb	22.7 kb	13.69 %

5. Conclusion

Arithmetic rules have been applied for compression for different kind of image is proposed here. Dividing the data used for storing the image or displaying is the best. It separate the data into two part 0(zero) to 9(nine) and 0(zero) to 25(twenty five). At the first stage it easily compress the value from 0(zero) to 255(two hundred fifty five) to 0(zero) to 25(twenty five), it includes0 (zero) to 9(nine). It shows a satisfactory result on the facial image, scenario image and algorithms efficient work on 10(ten) KB to 30(twenty) KB size of image. These algorithms are implemented by using C-programming language. In near future 'JPEG' image will be considered as test image of this algorithms and further work is going on to research for more efficient compression algorithm for images.

References

Book, book chapter and manual

[1] Data Compression: The Complete Reference by D. Salomon, Springer Verlag, 2000.
[2] A Guide to Data Compression Methods (Paperback) by D. Salomon, Springer Verlag, 2002.
[3] Multimedia Systems.pdf by Ralf Steinmetz, KlaraNahrstedt – 2004
[4] Data Compression for Real Programmers by Peter Wayner
[5] LZW based compression by Chang Hong Lin, Yuan Xie, Wayne Wolf
[6] A short introduction from 2000 by Laurens Leurs of RLE.

Journals

[7] Nigar S Sulthana and Mahesh Chandra. Article: Image Compression with Adaptive Arithmetic Coding. International Journal of Computer Applications
[8] A new lossless method of image compression and decompression using Huffman coding techniques by JagdishH.Pujar,Lohit M.Kadlaskar,JATIT-2005.
[9] A STUDY OF VARIOUS IMAGE COMPRESSION TECHNIQUES by Sonal, Dinesh Kumar Department of Computer Science & Engineering Guru Jhambheswar University of Science and Technology, Hisar.
[10] Digital Preservation Guidance Note 5: Image compression, Author: Adrian Brown, Digital Archives Analyst. Document Reference DPGN-05 Issue Date: 9 July 2003
[11] A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees, by Amir Said and William A. Pearlman, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 243-250, June 1996.
[12] An Image Multiresolution Representation for Lossless and Lossy Image Compression, by Amir Said and William A. Pearlman, first submitted July 1994, and published in IEEE Transactions on Image Processing, vol. 5, pp. 1303-1310, Sept. 1996.
[13] Reversible Image compression via multiresolution representation and predictive coding," by Amir Said and William A. Pearlman, presented at the SPIE VCIP Symposium, Cambridge, MA, Nov. 1993