ISSN No. 0976-5697

Volume 7, No. 6(Special Issue), November 2016

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Nobel Approach for Image Compression

Gagandeep Kaur
Assistant Professor
Guru Nanak College,
Budhlada,Mansa,India
gagan.manshahia@gmail.com

Narinder Singh
Assistant Professor
Guru Nanak College Budhlada,
Mansa, India.
ns_kalra@yahoo.co.in

Sukhpreet Singh
Assistant Professor
Krishna college,Ralli, Budhlada
sukhnav.preet@gmail.com

Abstract-Image Compression is a technique that is used to reduce the size of an image by reducing redundancy in that image. Image compression techniques save the disk space to store an image and bandwidth. A Compressed Image takes less time to process than an image of more size. Image compression technique is applied to both lossy as well as lossless data. In this paper Entropy algorithm is compared with color correlation algorithm. In latest algorithm DCT with color co-relation technique is used in which firstly image is converted in to byte array. There is a loss of data while using Entropy algorithm but Latest algorithm overcomes the disadvantage of Entropy algorithm. There is no loss of data after compression of an image till 50%.

Keywords: Image compression, Color correlation, DCT, Entropy.

I. INTRODUCTION

An important role is played by Digital Image Processing despite the rapid progress in digital communications. Compressed Images takes less memory space [1]. In applications like video streaming, satellite imaging, medical imaging and high quality (High resolution) photography the size of data to be transferred is incredibly large when compared to the bandwidth available in the communication channel. Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements. It decreases the cost of storage requirement. The efficient compression techniques will continue to be a design challenge for future communication systems. Data is represented as a combination of information and redundancy [1]. Data compression can be understood as a method that takes an input data D and generates a shorter representation of the data c(D) with less number of bits compared to that of D. The reverse process is called decompression, which takes the compressed data c(D) and generates or reconstructs the data. Sometimes the compression (coding) and decompression (decoding) systems together are called a “C6ODEC”[1].

The reconstructed data D could be identical to the original data D or it could be an approximation of the original data D, depending on the reconstruction requirements. After Decompression [14], if the original size of an image is formed then this is lossless Compression technique. On the other hand, the Techniques are lossy when Decompression is not an exact replica of D. Hence the algorithms can be classified in two categories –

- Lossless
- Lossy.

Usually lossless [4] data compression techniques are applied on text data or scientific data. Noise can also treat as Lossy because noise means something disturbed. So no original data is formed if there is noise in the system [13].

II. DATA COMPRESSION MODEL

A model of a typical data compression system [4] can be described using the block diagram shown in Fig. 2. A data compression system mainly consists of three major steps – Removal or reduction in data redundancy, Reduction in entropy, Entropy encoding [1]. The redundancy in data may appear in different forms. For example, the neighboring pixels in an image are very much spatially correlated to each other. Correlation [4] means that the pixel values are very similar in the non-edge smooth regions in the image. Record can be similar means redundancy of data [2].

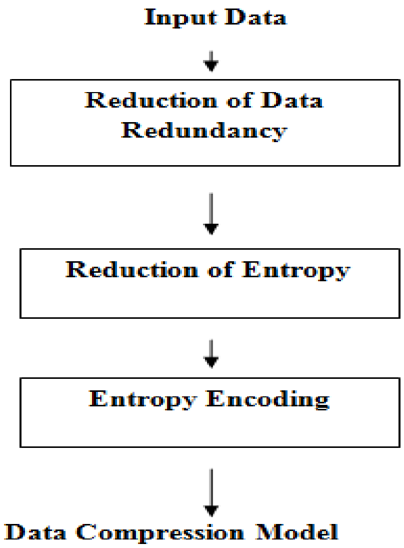


Fig. 1. Data Compressed Model

These redundancies in data representation can be reduced in order to achieve potential compression. In lossy [14] data compression system entropy [1]of the transformed data is reduced significantly in order to allocate fewer bits for transmission or storage. The reduction in entropy is achieved by dropping non significant information in the transformed data based on the application criteria. This is a nonreversible process because loss of data is not recovered. it can be represented by fewer bits compared to the original data set, hence compression is accomplished. Entropy[12] is the measure of the amount of uncertainty or information in the data. The larger the uncertainty of a random variable the larger is the entropy. Let *I* represent the self information of a random variable *ak*, whose **P** (a_k) is the probability of *ak*. Then

- $I(a_k) = -\log P(a_k)$ (1)
- If P (a_k) = 1 then I = 0 (2)

if it is certain that an event is going to happen then the information is that event is zero.

-
- Where **P** (a_k) is the probability that the symbol (a_k) in S will occur[1].

III. EFFECTS OF IMAGE COMPRESSION

- **COMPRESSION ARTIFACT**

Artifacts are often a result of the latent errors inherent in lossy data compression. Some of the common artifacts are:

- *Blocking Artifacts* : A distortion that appears in compressed image[9] as abnormally large pixel blocks. Also called "macroblocking," it occurs when the encoder cannot keep up with the allocated bandwidth[12]
- *Color Distortion*: As human eyes are not as sensitive to color as to brightness, much of the detailed color (chrominance) information is disposed, while luminance is retained. This process is called "chroma sub-sampling", and it means that a color image is split into a brightness image and two color images.
- *Ringing Artifacts*: In digital image processing, ringing artifacts are artifacts that appear as spurious signals ("rings") near sharp transitions in a signal.
- *Blurring Artifact*: Blurring means that the image is smoother than originally [5].

IV. SOURCE CODING ALGORITHM

Source coding can mean both lossless and lossy compression [5]. Depending on the characteristics of the data, each algorithm may give different compression [13] performance. So selection of the particular algorithm will depend upon the characteristics of the data themselves. In a lossy compression mode, the source coding algorithms are usually applied in the entropy encoding step after transformation and quantization.

A. Run-Length Coding

Run-length coding is a simple approach to source coding when there exists a long run of the same data, in a consecutive manner, in a data set. As an example, the data *d* = 5 5 5 5 5 5 5 19 19 19 19 19 19 19 19 19 19 19 19 0 0 0 0 0 0 0 0 23 23 23 23 23 23 contains long runs of 5‟s, 19‟s, 0‟s, 23‟s etc. Rather than coding each sample in the run individually, the data can be represented compactly by simply indicating the value of the sample and the length of its run when it appears. In this manner the data *d* can be run-length encoded as (5 7) (19 12) (0 8) (23 6). Here the first value represents the pixel, while the second indicates the length of its run[5].

B. Huffman Coding

Huffman [13] invented a coding technique to produce the shortest possible average code length given the source symbol set and the associated probability of occurrence of the symbols. Codes generated using these coding techniques are popularly known as Huffman codes. Huffman coding technique is based on the following two observations regarding optimum prefix codes.

- The more frequently occurring symbols can be allocated with shorter code words than the less frequently occurring symbols.
- The two least frequently occurring symbols will have code words of the same length, and they differ only in the least significant bit.

C. Arithmetic coding

In arithmetic coding a sequence of input symbols is represented by an interval of real numbers between 0.0 and 1.0. Arithmetic coding offers superior efficiency and more flexibility compared to the popular Huffman coding [5]. It is particularly useful when dealing with sources with small alphabets such as binary alphabets and alphabets with highly skewed probabilities [10].

Limitations of technique

- Arithmetic code needs higher computations.
- It is not accurate for converting the decompression data into original.

D. Encoding Algorithm

The arithmetic coding algorithm is explained here with an example [1]. Consider a four-symbol alphabet $A = \{a, b, c, d\}$ with the fixed symbol probabilities $p(a) = 0.3$, $p(b) = 0.2$, $p(c) = 0.4$, and $p(d) = 0.1$ respectively. The symbol probabilities can be expressed in terms of partition of the half-open range [0.0, 1.0)]. Entropy encoder [12] algorithm works on quantized coefficients of the discrete cosine transform. The probability of being nonzero of zig-zag ordered DCT [11] coefficients, is a decreasing monotonic function of the index. This is shown in Fig. 3.

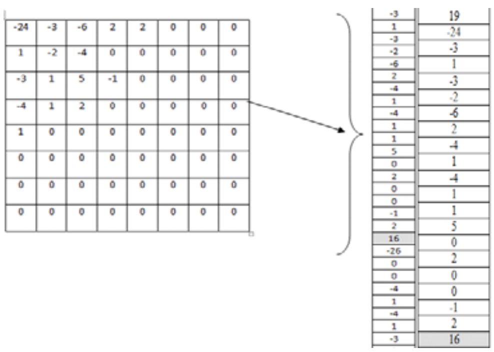


Fig. 2 Image conversion in to Bytes

Entropy encoder algorithm works on quantized coefficients of the discrete cosine transform. The probability of being nonzero of zig-zag ordered DCT[5] coefficient, is a decreasing monotonic function of the index.

V. PROBLEM DEFINITION

As an Enhancement in technology is going on, the size of everything like pan drive, Computer, CD, DVD is reduced. To store the large content in database is typical rather than to store a bits of information. If the size of contents is reduced without missing the original contents, it is easy to save the space where we want to store the contents. It improves the time period to access a data. For this purpose Image compression techniques are used. The problem definition is to compress an image using color correlation technique.

VI. OBJECTIVES

- Compress the image more than 50% of the size with no compromise of quality standard with loss of image or artefact.
- Bandwidth of internet speed may increase.
- Need less storage capacity.
- Easy to store on flash memory used by flash drive, JPEG images, players.
- Time saving on uploading & downloading of such images.

VII. METHODOLOGY

The basic idea of the new approach is to divide the image into 8x8 blocks and then extract the consecutive non-zero coefficients preceding the zero coefficients in each block. In contrast to the Run Length decoder, the output of this encoder consists of the number of the non-zero coefficients followed by the coefficients themselves for each block. The decompression process can be performed systematically and the number of zero coefficients can be computed by subtracting the number of non-zero coefficients from 64 for each block. Following is a short example of this algorithm. In this example, there are two 8x8 blocks of coefficients as input to the suggested coder. The output of this coder is shown in the right side of Fig given above.

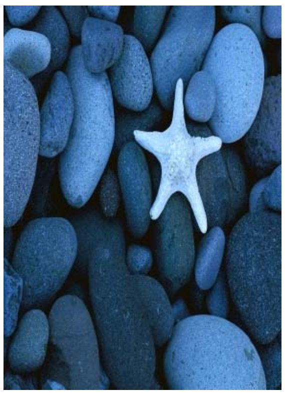
But in our Image de-composition & color correlation Method, Image is firstly converted into byte[] stream and match the values of color by using binary tree to sort the values of color and mark the reference of the location of stream index. This index will fill the color of the reference location. This helps to no reduction of value to achieve the lossless image. There is no neighbor color co relation by which more efficient image compression will achieved.

Algorithm used in image compression

Step1:- Browse a Digital image.
Step2:- Convert digital image into small pixel and make an array list.
Step3:- Apply D Cosine Transformation conversion.
Step4:- Fetch value of red(R) color mode of image.
Step5:- Replace all same value with references of an array.
Step6:- Repeat same above for Green and Blue.
Step7:- Write image to the hard-disk with following steps.

```
String fn;
fn=imagepath.image;
    FileStream fs=new FileStream(fn,FileMode.Create,FileAccess.Write);
byte[] ar=(Byte[])imgnam;
fs.Write(ar,0,ar.Length-1)
```

1.Chrysanthemum



2. Sea star

3. Desert

4.Flowers

5 .Penguins

We have worked on the following images as shown above. On compression the images size is reduced as remands change in size in such a way:-

Table1 shows the size of images after compression

Sr. No.	Image Name	Frame size	New Size
•	Chrysanthemum	858kb	115 kb
•	Sea Star	834kb	166 kb
•	Desert	826kb	119 kb
•	Flower	760kb	361 kb
•	Penguins	891kb	102kb

Table 1.Size Comparison

The size reduced more than 50%. As mathematically or by the result if movers that the effective as compare to the entropy method.

VIII. CONCLUSION AND FUTURE WORK

In our image compression technique, we user bit reference so there is no way to effect to image for any type of generation of image artifacts. It not only reduces storage requirements but also overall execution time and improves the performance. In future As we focused on the DCT value and RGB on the pixel may vary and remove the save value of RGB on the image But the value of the pixel may vary from image to image as per the design and structure of image. So if the image has differently colored used or we can say that the pixel value are unique. In the image then image must be need some other enhancement. This may future work on this type of images.

IX REFERENCES

[1] M. I. Khalil , "Image Compression Using New Entropy Coder" International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010.

[2] Risto Miikkulainen, "Effective Image Compression using Evolved Wavelets" Department of Computer Sciences, The University of Texas at Austin.

[3] S.R. Kodituwaku & U.S. Amarasinghe Compression of Lossless Data Compression Algorithms For Text Data, S.R. Kodituwakku et. al. / Indian Journal of Computer Science and Engineering, Vol 1 No 4 416-425.

[4] Xing San , Hua Cai t and Jiang Li, Color Image Coding By Using Inter-Color Coralaton, Dept. of Electronic Eng.
and Information Science, University of Science and Technology of Chinat Media Communication Group, Microsoft Research Asia, Beijing, China

[5] Sonal, Dinesh Kumar , A Study Of Various Image Compression Techniques , Department of Computer Science & Engineering Guru Jhambheswar University of Science and Technology, Hisar

[6] D. S. Taubman, High performance scalable image compression with EBCOT, IEEE Transaction Image Processing, Vol. 9, No. 7, pp. 1158–1170, July 2000

[7] JPEG 2000 Final Committee Draft. Boliek, M. 2000.

[8] Evaluation of Design Alternatives for the 2-D-Discrete Wavelet Transform.

[9] A. Skodras, C. Christopoulos, and T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine, pp. 36-58, September 2001.

[10] P. Chen, T. Acharya, and H. Jafarkhani, A Pipelined VLSI Architecture for Adaptive Image Compression, International Journal of Robotics and Automation, Vol. 14, No.3, pp. 115-123, 1999.

[11] Christian Piguet, Taylor, Francis (2006), Low Power CMOS Circuits, Technology, Logic Design and CAD Tools by -2006.

[12] Maddu Karunaratne, Chamara Ranasinghe (2008), A Dynamic Switching Activity Generation Technique for Power Analysis of Electronic Circuits, 48th Midwest Symposium on Circuits and Systems.

[13] E. Atsumi and N. Farvardin, Loss/Lossless Region-of-Interest Image Coding Based on Set Partitioning in Hierarchical Tree, IEEE International Conference Image Processing, pp. 87-91, Chicago, October 1998.

[14] D. Nister and C. Christopoulos, Lossless Region of Interest with Embedded Wavelet Image Coding, Signal Processing, Vol. 78, No.1, pp. 1-17, 1999.