OPEN ACCESS
International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-3, February 2013

Image Compression Method

Sunanda Kisanrao Kapde, S.V.Patil,

***Abstract-* Data transmission and storage cost money. The more information being dealt with, the more it costs. In spite of this, most digital data are not stored in the most compact form. Rather, they are stored in whatever way makes them easiest to use. Data compression is the general term for the various algorithms and programs developed to address this problem. A compression program is used to convert data from an easy-to-use format to one optimized for compactness. Here two algorithms were selected namely, the original block truncation coding (BTC) and Absolute Moment block truncation coding (AMBTC) and a comparative study was performed. The results have shown that the ATBTC algorithm outperforms the BTC. It has been show that the image compression using AMBTC provides better image quality than image compression using BTC at the same bit rate. Moreover, the AMBTC is quite faster compared to BTC.**

***Index Terms:* BTC, AMBTC, Q level quantizer, image compression; mean, standard deviation**

I. INTRODUCTION

The amount of image data grows day by day. Large storage and bandwidth are needed to store and transmit the images, which is quite costly. Hence methods to compress the image data are essentially now-a-days. The image compression techniques are categorized into two main classifications namely Lossy compression techniques and Lossless compression techniques. Lossless compression ratio gives good quality of compressed images, but yields only less compression whereas the lossy compression techniques lead to loss of data with higher compression ratio. [1] Data Compression shrinks down a file so that it takes up less space. This is desirable for data storage and data communication. Storage space on disks is expensive so a file which occupies less disk space is "cheaper" than an uncompressed file. Smaller files are also desirable for data communication, because the smaller a file the faster it can be transferred. A compressed file appears to increase the speed of data transfer over an uncompressed file.

The performance of any compression technique is measured in terms of 1.Compression ratio 2.Signal-to-Noise Ratio (quality of the reconstructed image) and 3.Encoding and Decoding Speed. In lossy image compression method, the redundant information can be compressed. One such lossy image compression technique is Block Truncation coding (BTC) technique. BTC is one of the recent techniques used to compress image data. The computational complexity is lee in terms of BTC when compared to other techniques.

Manuscript published on 28 February 2013.
***** Correspondence Author (s)
Sunanda Kisanrao Kapde, Lecturer, E & TC Department, B.l.Patil Polytechnic Khopoli,Maharashtra,India
Mr.S.V.Patil, Associate professor, E & TC Department, J.T.Mahajan college of Engg,Faizpur Maharashtra,India.

A compression rate of 2 bpp is achieved with BTC . The compressed image is only an approximation of the original image. Though some data is removed, for Human Visual System, there is no much difference between the original and the compressed image. In encoding process, for each input image block of size 4x4 pixels, a bit plane is formed by replacing each pixel value with either 1 or 0. When the pixel value is greater than mean, it is coded as 1, otherwise 0. Two statistical moments of size 16 bits are preserved along with the bit plane.

The bit-rate achieved with BTC can further be reduced by reducing the amount of data needed to represent the Bit plane by dropping few bits. The dropped bits can be generated at the decoding stage by using the Interpolation technique. By using interpolation method, the size of bit plane is reduced from 16 bits to just 8 bits leading to a bi trate of 1.75 bpp reduced from 2 bpp. The total number of bits is 24 for each block and the bit rate is reduced from 2 to 1.5 bits/pixel. In BTC, the statistical overheads: the standard Mean and standard deviation are preserved for each input block. The truncated block of the BTC is the one-bit output of the quantizer for every pixel in the block. A simple variant of BTC called Absolute Moment Block Truncation Coding (AMBTC) was presented by Lema and Mitchell, and it preserves the higher mean and lower mean of the blocks . The interpolative techniques are used to code the compressed bit plane of BTC. An algorithm for preserving moments which results in less mean square error (MSE) was proposed in [2]

II. STANDARD BTC ALGORITHM

The basic BTC algorithm is a lossy fixed length compression method that uses a Q level quantizer to quantize a local region of the image. The quantizer levels are chosen such that a number of the moments of a local region in the image are preserved in the quantized output. In its simplest form, the objective of BTC is to preserve the sample mean and sample standard deviation of a grayscale image. Additional constraints can be added to preserve higher order moments. For this reason BTC is a block adaptive moment preserving quantizer

The first step of the algorithm is to divide the image into non-overlapping rectangular regions. For the sake of simplicity we let the blocks be square regions of size n x n, where n is typically 4. For a two level (1 bit) quantizer, the idea is to select two luminance values to represent each pixel in the block. These values are chosen such that the sample mean and standard deviation of the reconstructed block are identical to those of the original block. An n x n bit map is then used to determine weather a pixel luminance value is above or below a certain threshold.

2.1. The BTC algorithm

Involves the following steps:-

For compression:-

1. Input image
2. Set block size
3. Calculate vertical and horizontal blocks
4. Create matrix (1) for saving means, std of blocks. Binary matrix and comp
5. For each block
   - Calculate and save mean
   - Calculate and save std.
   - Get binary matrix by comparing pixel value with mean
6. Compress binary matrix for mean and std. in above step(considering 8 binary value = 1 byte)
7. Save means, stds, compressed mat data.

For decompression:-

Load save d data

2) Decompress compressed matrix (1 byte = 8 binary values)
3) For each block in decompressed mat
   - Extract block
   - Get mean and std of block
   - Calculate no. of ones in block (say 9)
   - Calculate values of a and b
   - Replace 0 by a and 1 by b in block
4) Show decompressed mat.

An example is shown in following tables:

The original image block is-

The average gray level = 159

181	173	156	141
192	180	138	118
188	183	141	116
185	176	157	123

The encoded information: the binary block (16 bits),

$m_u = 182, m_l = 136$

.

Total = 32 bits/block.

\1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0

The reconstructed image block by the decoder is-

182	182	136	136
122	182	136	136
182	182	136	136
182	182	136	136

The error (difference between the original and the reconstructed image block) is-

Mean square error = 128.5.

-1	-9	20	5
10	-2	2	-18
6	1	5	-20
3	-6	21	-13

2.2 ALGORITHM OF AMBTC

Lema and Mitchell presented a simple and fast variant of BTC, named Absolute Moment BTC (AMBTC) that preserves the higher mean and lower mean of a block. The AMBTC algorithm involves the following steps:

• Step 1:

An image is divided into non-overlapping blocks. The size of a block could be (4 x 4) or (8 x 8), etc.

• Step 2:

Calculate the average gray level of the block (4x4) as equations x_H

• Step 3:

Pixels in the image block are then classified into two ranges of values. The upper range is those gray levels $\overline{x}$ which are greater than the block average gray level and the remaining brought into the lower range. The mean of higher range XH and the lower range XL are calculated as:

$$x_H = \frac{1}{K} \sum_{x_i > \overline{x}}^{n} x_i$$

$$x_L = \frac{1}{16 - K} \sum_{x_i < \overline{x}}^{n} x_i$$

Here k is the number of pixels whose gray level is greater than

$\overline{x}$

Step 4:

Binary block, denoted by B, is also used to represent the pixels. We can use “1” to represent a pixel whose gray level is grater than or equal to an $\overline{x}$ d “0” to represent a pixel whose gray level is less than Th $\overline{x}$ e encoder writes XH , XL. Then the total number of bits required for a block is 8+8+16 =32 bits. Thus, the bit rate for the AMBTC algorithm is 2 bpp

$$B = \begin{cases} 1 & x_i \geq \overline{x} \\ 0 & x_i \prec \overline{x} \end{cases}$$

Step 5:

In the decoder, an image block is reconstructed by replacing the `1' s with XH and the '0''s by XL In the AMBTC, we need 16 bits to code the bit plane which is same as in the BTC. But, AMBTC requires less computation than BTC

$$X = \begin{cases} X_L & B = 0 \\ X_H & B = 1 \end{cases}$$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

AMBTC has several advantages over BTC one advantage is in the case that the quantizer is used to transmit an image from transmitter to a receiver, it is necessary to compute at the transmitter the two quantities, the sample mean and the sample standard deviation for BTC and sample first absolute central moment for AMBTC. When we compare the necessary computation for deviation information, we will see that in case of standard BTC it is necessary to compute a sum of m values and each of them will be squared while in case of AMBTC it is only necessary to compute the sum of these m values. Since the multiplication time is several times greater than the addition time in most digital processors, thus using AMBTC the total calculation time at the transmitter is significantly reduced.[1]

An example for AMBTC encoding procedures are shown in Figure 1. Figure 1 (a) shows an image block of 4×4 pixels. To encode this block, the mean of this block is calculate as o   =225 . The value o is then taken as a threshold to generate a bit plane B, as shown in Figure 1(b). Pixels with values higher than or equal to o have a corresponding bit valued 1 stored in the bit plane. Otherwise, bit valued 0 is stored. Two rounded quantization levels a =223 and b = 226 can be calculated using Equations (1) and (2). Once the quantization levels and the bit plane have been calculated, the compressed code of this image blocks comes out as trio of [223,226,(1100 1100 1100 1100)]. All image blocks are encoded with the same manner, until all the blocks are processed

226	226	223	223
226	226	223	223
226	226	223	223
226	226	223	223

(a) Original image block image

1	1	0	0
1	1	0	0
1	1	0	0
1	1	0	0

(b) AMBTC bit plane

226	226	223	223
226	226	223	223
226	226	223	223
226	226	223	223

(c) Reconstructed image blocks

Example of AMBTC encoding and decoding procedures

To decode an AMBTC-compressed image, the decoder reconstructs image blocks from the compressed code. To recover an image block, if a value b i in B is 0, then the corresponding pixel is reconstructed by the quantization level a, Otherwise, reconstructed by the quantization level b. By recovering pixels in all image blocks, the whole compressed Image can be reconstructed. [3]

III. RESULTS AND DISCUSSION

For the result we select hard disc image, to compare between the used compression methods, comparison is a follows:-

Original Size of the stone is 16384 pixels.

Fig 1: Original image

Fig 2a: BTC Compressed image Compressed Size 2128 Compression Ratio 4.923077BTC Bit size = 8

Fig 2b: AMBTC Compressed image Compressed Size 2816 Compression Ratio 5.818182 AMBTC Bit size = 8

Fig3a: BTC Compressed image Compressed Size 2128 Compression Ratio 6.918919 BTC Bit size = 16

Fig 3b: AMBTC Compressed image Compressed Size 2240Compression Ratio 7.314286 AMBTC Bit size = 16

Fig 4a: BTC Compressed image Compressed Size 2128 Compression Ratio 7.699248 BTC Bit size = 32

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

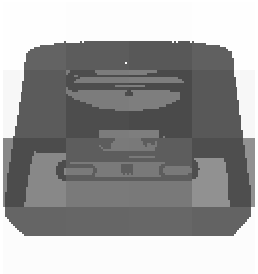

Fig4b: AMBTC Compressed image Compressed Size 2096 Compression Ratio 7.816794 AMBTC Bit size = 32

It is observed that, as compression ratio increases, decompressed hard disk image, contain less detail block of hard disk is also increases. For the same block size compression ratio of AMBTC image is always greater than BTC image. Here for the block size16, compression ratio for uncompressed hard disk image is 4.923077 and for compressed image 5.818182. Also for the block size8, compression ratio for uncompressed hard disk image is and for compressed image 7.314286.

IV. CONCLUSIONS

In this paper, image compression using block truncation coding has been investigated. Two algorithms were selected namely, the original block truncation coding (BTC) and Absolute Moment block truncation coding (AMBTC). The two algorithms are based on dividing the image into non overlapping blocks and uses a two-level quantize. .It has been show that the image compression using AMBTC provides better image quality than image compression using BTC at the same bit rate. Moreover, the AMBTC is quite faster compared to BTC

REFERENCES

[1] Doaa Mohammed, Fatma Abou-Chadi," Image Compression Using Block Truncation Coding Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), February Edition, 2011.

[2] S. Vimala, M. Sathya , K. Kowsalya Devi ," Enhanced AMBTC for Image Compression using Block Classification and Interpolation" *International Journal of Computer Applications (0975 – 8887) Volume 51– No.20, August 2012*

[3] Edward J. Delp, Martha Saenz, and Paul Salama BLOCK TRUNCATION CODING (BTC)"Video and Image Processing Laboratory School of Electrical and Computer Engineering Purdue University West Lafayette, Indiana

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)