

Image Compression using Absolute Moment Block Truncation Coding

Shashikumar.S⁽¹⁾, Arpana Parakale⁽²⁾, Bharamgonda Madhuri Mahavir⁽³⁾,
Bharamu Ullagaddi⁽⁴⁾

Department of Electronics & Communication Engineering
Bahubali College of Engineering, Shravanabelagola, Hassan Dist

Abstract

The present work investigates image compression. Image compression is concerned with minimization of the number of information carrying units used to represent an image. With the continuing growth of modern communications technology, demand for image transmission and storage is increasing rapidly. The number of bits required to represent an image may be reduced considerably by coding techniques. In this paper Absolute Block Truncation Coding (AMBTC) technique is used for image compression. It is a lossy image compression algorithm, which is simple and easy to implement. AMBTC algorithm is a lossy fixed length compression method that uses a Q level quantizer to quantize a given region of the image. This technique rely on applying divided image into non overlapping blocks.

Keywords: Block Truncation Coding, AMBTC, Q level quantizer, image compression; mean, standard deviation

I. INTRODUCTION

Image Compression shrinks down a file so that it takes up less space. The amount of data required to represent a digital image is more, by using a suitable image compression technique this can be reduced. It is necessary to find efficient representation for digital images in order to reduce the memory required for storage, to improve the data access rate from storage devices and to reduce the bandwidth and time required for transfer across communication channels. Figure 1 gives an insight on the different compression algorithms developed in the literature.

Block Truncation Coding (BTC) technique also called a moment preserving quantizer. It is a image compression scheme that was proposed by Delp and Mitchell. In BTC an image is firstly segmented into pixels of $n \times n$ blocks. Secondly, a two level output is chosen for every block and is encoded using a 1 bit for every pixel. Two levels are selected independently for every $n \times n$ blocks, so encoding the block is a simple binarization process, because every pixel of the block is truncated to either a 1 or a 0 binary representation choosing the reconstruction level (the two level output). The quantizer threshold and the two reconstruction levels are varied in response to the local statistics of a block. As BTC requires square root operations for its implementation, a more efficient algorithm, AMBTC has been extensively used in signal compression because of its simple computation and better Mean Square Error (MSE) performance.

Lema and Mitchell have designed their one-bit BTC by maintaining the first absolute central moment (or average deviation), besides the sample mean, leading to an AMBTC. The resulting quantizer has a simpler algorithmic structure and smaller computational errors compared with that of original BTC [2]. AMBTC has the same general characteristics as BTC which include low storage requirements and an extremely simple coding and decoding technique. The central idea of AMBTC is to preserve higher mean and lower mean of each small rectangular block of pixels spatially and nonoverlappingly divided from the original image. Thus it is necessary to increase the number of quantization levels to improve the image quality.

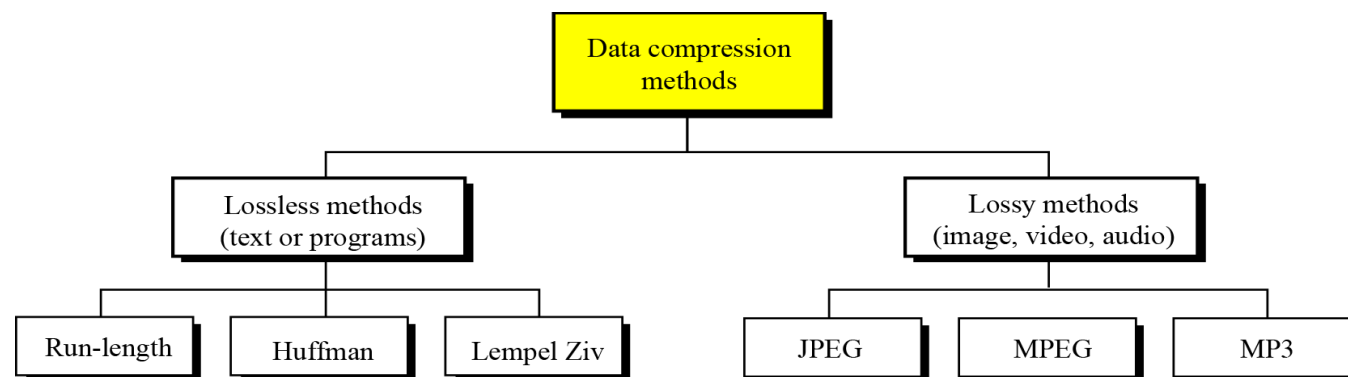


Fig. 1: Types of Data Compression Methods

The rest of the paper is organized as follows: Section II deals with the types of image compressions. Sections III and IV detail BTC and AMBTC algorithms respectively. Section V gives the simulation results and analysis. Finally section VI gives the concluding remarks.

II. TYPES OF IMAGE COMPRESSION

Image compression implies sending or storing a smaller number of bits. Although many methods are used for this purpose, in general these methods can be divided into two broad categories: lossless and lossy methods.

In lossless data compression, the integrity of the data is preserved. The original data and the data after reconstruction are exactly the same because, in these methods, the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process. Redundant data is removed in compression and exactly an inverse process is carried out during decompression. Lossless compression methods are normally used when we cannot afford to lose any data.

If lossy compression is applied to an image, the image cannot be recovered exactly as it was before compression. When the compressed image is decoded it does not give back the original image. Data has been lost because lossy compression cannot be decoded to yield the exact original image. Hence it is not a good method of compression for critical data, such as textual data. It is most useful for Digitally Sampled Analog Data (DSAD). DSAD consists mostly of sound, video, graphics, or picture files.

Our eyes and ears cannot distinguish subtle changes. In such cases, we can use a lossy data compression method. These methods are cheaper—they take less time and space when it comes to sending millions of bits per second for images and video. Several methods have been developed using lossy compression techniques. JPEG (Joint Photographic Experts Group) encoding is used to compress pictures and graphics, MPEG (Moving Picture Experts Group) encoding is used to compress video, and MP3 (MPEG audio layer 3) for audio compression.

Figure 2 represents the block diagram of image compression, in which $f(x, y)$ is a digital image. It is compressed using a suitable compression algorithm. Then it is transmitted over a channel to the receiver end. At the receiver end the data is retrieved from the storage device for decompression. The decompressed data will yield $\hat{f}(x, y)$. Here $f(x, y)$ is original image and $\hat{f}(x, y)$ is the reconstructed image.

III. STANDARD BTC ALGORITHM

BTC works by dividing the image into small sub blocks of size $n \times n$ pixels and then reducing the number of gray levels within each block. BTC divides the whole image into N blocks and codes each block using a two-level quantizer. The two levels, a and b are selected using the mean (\bar{X}) and standard deviation (σ) of the gray levels within the block and are preserved. Each pixel value within the block is then compared with the mean and then is assigned to one of the two levels [1, 2]. The \bar{X} and σ are calculated using the equations (1) and (2).

$$\bar{X} = \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\sigma = \sqrt{\frac{\sum (y_i - x_i)^2}{m}} \quad (2)$$

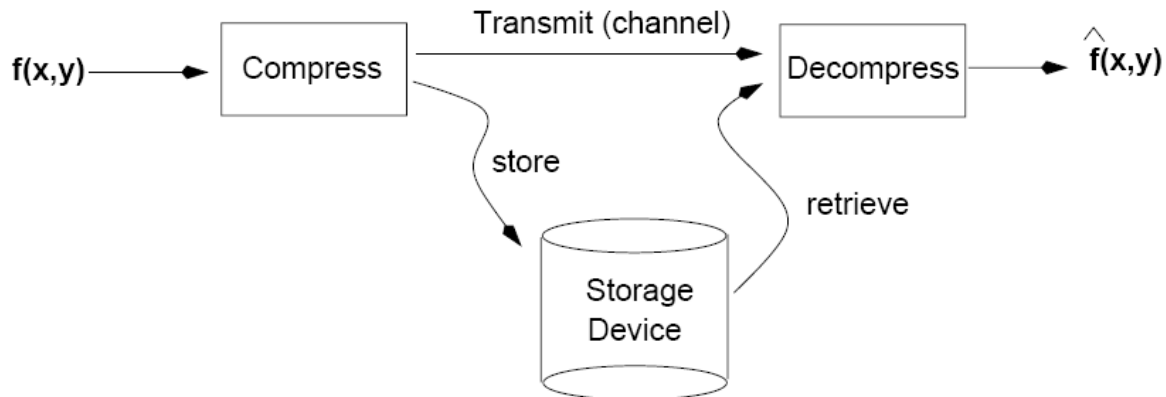


Fig. 2: Block Diagram of Image Compression

Where, m is the number of pixels in each block, and X_i is the original pixel value of the block. If the pixel value of each block is greater than or equal to mean, it is represented by “1” and if less than the mean, it is represented by “0”. The collection of 1s or 0s for each block is called a bit plane, B . In BTC, two Statistical moments a and b are computed using the equations (3) and (4) and are preserved along with the bit plane for reconstructing the image. The compressed image is transmitted or stored as a set $\{B, a, b\}$

$$a = \bar{X} - \sigma \sqrt{\frac{q}{m-q}} \quad (3)$$

$$b = \bar{X} + \sigma \sqrt{\frac{m-q}{q}} \quad (4)$$

Where, q is the number of pixel values greater than or equal to \bar{X} and $(m-q)$ is the number of pixels whose gray levels are less than \bar{X} . While reconstructing the image, the 0 in the bit plane is replaced by a and the 1 in the bit plane is replaced by b . The difference between the original image and reconstructed image is called MSE and is calculated using the equation (5). The quality of the reconstructed image called the Peak Signal to Noise Ratio (PSNR) is calculated using the equation (6) and is the inverse of MSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - x_i)^2 \quad (5)$$

$$PSNR = 10 \cdot \log_{10} [MSE/255^2] \quad (6)$$

Where, y_i is the reconstructed pixel value, x_i is the original pixel value and N is the number of pixels in an image.

IV. ABSOLUTE MOMENT BLOCK TRUNCATION CODING

Lema and Mitchell presented a simple and fast variant of BTC named AMBTC. In this method, the higher mean h_x and lower mean l_x are preserved instead of the mean and standard deviation values. Pixels in an image block are then classified into two groups of values. One group (higher range) comprising of gray levels which are greater than or equal to the mean (\bar{x}) and the remaining gray levels are brought into another group (lower range). The mean h_x of higher range and l_x of the lower range are calculated using the Eq. (7) and Eq. (8).

$$\overline{x_l} = \frac{1}{(m-q)} \sum_{x_i < x} x_i \quad (7)$$

$$\overline{x_h} = \frac{1}{(q)} \sum_{x_i > x} x_i \quad (8)$$

Where, q is the number of pixels whose gray levels are greater than or equal to x. A two level quantization is performed for all the pixels in that block to form a bit plane of 1's and 0's. Encoder generates l_x , h_x and bit plane for each block, thus leading to 2 bpp like BTC. AMBTC involves less number of computations, compared to BTC as standard deviation involves more multiplications. AMBTC yields high quality of reconstructed images. [2].

V. RESULTS AND DISCUSSIONS

This section gives an insight on the performance of AMBTC for various values of mxn. An example for AMBTC encoding procedures are shown in Figure 3. Figure 3 (a) shows an image block of 4x4 pixels. To encode this block, the mean of this block is calculate as $\mu = 234$. The value μ is then taken as a threshold to generate a bit plane B, as shown in Figure 3(b). Pixels with values higher than or equal to μ have a corresponding bit valued 1 stored in the bit plane. Otherwise, bit valued 0 is stored.

Two rounded quantization levels $a = 228$ and $b = 243$ 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 [228, 243, (0001 1101 1001 0001)]. All image blocks are encoded with the same manner, until all the blocks are processed.

To decode an AMBTC-compressed image, the decoder reconstructs image blocks from the compressed code. To recover an image block, if a value b 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]

223	229	227	245
246	248	230	235
244	225	231	240
233	222	232	245

(a)

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

(b)

228	228	228	243
243	243	228	243
243	228	228	243
228	228	228	243

(c)

Fig 3: (a) Original image block (b) AMBTC bit plane (c) Reconstructed image block

Table 1: Compression Ratio,SNR,MSE,PSNR for different values of MxN

M	N	Compression Ratio	SNR	MSE	PSNR
4	4	4	10.1270	13.1285	36.9826
8	8	6.4	10.048	29.1442	33.5193
16	16	7.5294	9.9911	51.0365	31.0860
32	32	7.8769	9.7135	75.0892	29.4090

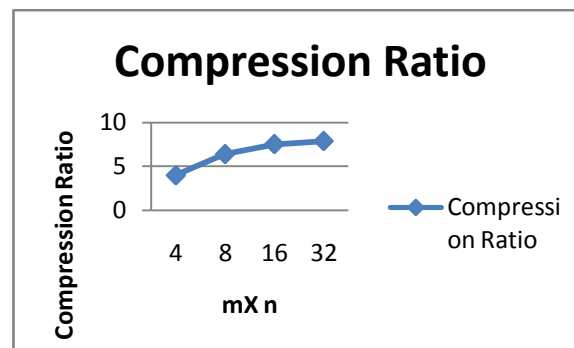


Fig. 4: Compression ratio v/s Block Value



Fig. 5: Original Image



Fig. 6: Compressed Image

The table1 shows the compression ratio for different value of MxN. The compression ratio achieved by AMBTC is approximately 8. As the MxN value increases the compression ratio also increases and reaches maximum upto 8.

Figure 4 shows the plot of compression ratio for different values of MxN values. From the above plot we can conclude that as the MxN value increases the compression ratio also increases and attains a maximum of 8. Figure 5 shows a typical example of an image as applied with AMBTC algorithm.

VI. CONCLUSIONS

In this paper, image compression using Absolute Moment Block Truncation Coding (AMBTC) has been investigated. The algorithm is based on dividing the image into non overlapping blocks and uses a two-level quantize. It has been showed 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.

ACKNOWLEDGEMENTS

We would like to take this opportunity to thank one and all who have provided their valuable advice, without their guidance this work would not have been a success, we have to thank who have helped us directly or indirectly since they have given us more than just guidance.

We want to thank to **Mr. Rizwan M Pasha**, Associate professor, Department of Electronics and Communication Engineering, Bahubali College of Engineering, Shravanabelagola, who have guided us in our work and enrich it with his faithful experience.

We want to thank to **Prof. K S Shivaprakasha**, Associate professor, Department of Electronics and Communication Engineering, Bahubali College of Engineering, Shravanabelagola, who have guided us in our work and enrich it with his faithful experience.

Our profound thanks to **Mrs. Ramamani K**, Head of the Department, Department of Electronics and Communication Engineering, Bahubali College of Engineering, Shravanabelagola, for her invaluable advice and constant encouragement to complete this work in a successful manner.

REFERENCES

- [1] Meftah M.Almrabet, Amer R. Zerek, Allaoua Chaoui and Ali A. Akash," Image Compression using Block Truncation Coding", IJ-STA, Vol 3, No.2, PP.1046-1053, December 2009
- [2] S.Vimala, P.Uma, B.Abidha "Improved Adaptive Block Truncation Coding for Image Compression" *International Journal of Computer Applications (0975 – 8887) Volume 19– No.7, April 2011*
- [3] 1D.Kiran Kumar , 1E.V.Narayana , ECE Department, JNT University ,Kakinada, "Lossless Steganography for AMBTC –Compressed Color Images"
- [4] Rafael C. Gonzalez, Richard Eugene; "Digital image processing", Edition 3, 2008, page 466
- [5] M. Ghanbari "Standard Codecs: Image Compression to Advanced Video Coding" Institution Electrical Engineers , ISBN: 0852967101, 2003 , CHM , 430 pages.