
***COLOR IMAGE COMPRESSION BASED ON ABSOLUTE MOMENT
BLOCK TRUNCATION CODING USING DELTA ENCODING AND
HUFFMAN CODING***

By

N.A.A.Mosa

Department of computer science

Mansoura University, 35516

Egypt

Research Journal Specific Education

Faculty of Specific Education

Mansoura University

ISSUE NO. 35, JULY. 2014

مجلة بحوث التربية النوعية – جامعة المنصورة

العدد الخامس والثلاثون – يوليو ٢٠١٤

COLOR IMAGE COMPRESSION BASED ON ABSOLUTE MOMENT BLOCK TRUNCATION CODING USING DELTA ENCODING AND HUFFMAN CODING

*N.A.A.Mosa**

Abstract

A simple and easy to implement technique for improving absolute moment block truncation coding (AMBTC) is proposed. AMBTC method produce a fixed length binary representation of each block according to the block size of the image. In contrast huffman coding produce a variable length binary representation based on the statistical nature of the image. So the proposed scheme presented to improve the performance of AMBTC to achieve variable length compression by applying the combination of delta encoding , and huffman coding. The Performance of the proposed scheme is compared with the original BTC , and AMBTC in terms of compression ratio (Cr), root mean square error (RMSE), and peak signal-to-noise ratio (PSNR).Simulation results indicate that the compression ratio of the proposed algorithm is much higher than that of the original BTC , and AMBTC with a relative distortion of image quality in the reconstructed images.

Keywords: Image compression; Absolute moment block truncation coding; Delta encoding ;Huffman coding.

1. Introduction

Image compression is the art or science of effectively coding digital images to reduce the number of bits required to represent the image [1].It reduces the size of the image, so that the compressed image could be sent over the computer network from one place to another in short amount of time. In addition, the compressed image helps to store more number of images on the storage devices [2].Image compression plays an important role in many applications, including image database, image communications, remote sensing , etc [3].There are different techniques for compressing images. They are classified into two classes called lossless and lossy compression techniques. In lossless compression the reconstructed

* Department of computer science, Mansoura University, 35516 ,Egypt

image is identical to the original image in every sense, whereas in lossy compression the reconstructed image is similar to the original image but not identical to it, because some image information is lost during compression process[4].

Block truncation coding (BTC) is a simple and fast lossy compression technique for digitized grayscale images originally introduced by Delp and Mitchell [5]. The main idea of BTC is to perform moment preserving (MP) quantization for blocks of pixels so that the quality of the image will remain acceptable and at the same time decrease the storage space [6]. Since BTC has the advantages including the encoding and decoding are extremely simple and fast[7], requires a smaller computational load and much less memory, produce sharper edges which is important for a human visual system [8], low computational cost, relatively high quality [9], easy to implement compared to other algorithm such as vector quantization, and transform coding[10]. It was widely used in many compression applications such as high definition television(HDTV), Internet video, digital cameras and printers [11],and software-based multimedia systems[8].On contrast the main drawbacks of original BTC are producing fixed length compression [11], achieving low compression ratio[9], and it doesn't perform well as well as transform coding such as JPEG[8].

A simple and fast variation of BTC, called absolute moment BTC (AMBTC) was presented by Lema and Mitchell.It preserves the higher and lower mean of a block [12]. AMBTC has some advantages over BTC such as providing better image quality [13],and the coding and decoding processes are very fast [6].

Delta encoding is a simple coding technique.The key feature is that the delta encoded signal has a lower amplitude than the original signal.On the other words, delta encoding has increased the probability that each sample's value will be near zero, and decreased the probability that it will be far from zero.So it used in image compression to improve the performamnce of entrpy coding techniques such as huffman coding[14].

It's well known that the huffman's algorithm is generating minimum redundancy codes compared to other algorithms. Huffman coding technique generating binary tree by calculating the probability value for each pixel in the image and sorting the pixels from the lowest probability value to the highest probability value.Then allocates zero to the left node and one to the right node starting from the root of the tree[4].

In this paper we propose an image compression scheme based on AMBTC by combining delta encoding and huffman coding to produce variable compression ratio of AMBTC. The remainder of this article is organized as follows: Section 2 introduces image compression and techniques. Section 3 discusses the proposed image compression technique. Section 4 includes the experimental results. Section 5 concludes this paper.

2. Image compression

A digital image is a discrete two-dimensional function , $f(x,y)$ of picture elements (pixels). A digital image can be presented as an $X*Y$ matrix where X refers to the number of image rows , and Y refers to the number of image columns [15].

There are three types of digital image:

Binary image (bi-level) where each pixel assumes one of only two discrete values: 1 or 0, where 1 is white , 0 is black [16].

Grayscale image (monochrome) which contains the values ranging from 0 and 255, where 0 is black , 255 is white and values in between are shades of grey [4].

Color image where each pixel of the image is represented by three color components, usually red, green and blue, shortly RGB. Each R, G and B is also in the range of 0 and 255 and each pixel is represented in three bytes. On the other hand a grayscale image is represented only by one byte, so that the storage space of color image is three times the size of grayscale image [4].

Image compression is the application of data compression on digital images[17]. The main principle of image compression is based on the fact that neighboring pixels are highly correlated ,where the adjacent pixels have the same color or very similar colors. This correlation is called spatial redundancy [18]. The purpose for image compression is to reduce the amount of data required for representing sampled digital images, and therefore reduce the cost for storage and transmission[3].

In the field of image compression , data redundancy plays an important role, where data with redundancy can be compressed; in contrast, data without any redundancy can not be compressed. So image compression techniques aims to reduce or remove the redundant data from the image [4]. There are three types of redundancies , as follows :

Coding Redundancy: which is presented when less than optimum (the smallest length) code words are used to represent image data [17]. There are several lossless techniques for constructs such a code i.e. Huffman coding, Arithmetic coding[3]

Interpixel Redundancy: which is concerned with the correlations between the pixels of an image[19]. There are several lossless techniques to reduce the interpixel redundancy , such as predictive coding and run-length coding [4].

Psychovisual Redundancy: which is due to data that is ignored by the human visual system(HVS) [17]. Quantization is the famous lossy method used to reduce psychovisual redundant from the image [4].

Figure 1 shows general model for image compression system. The first stage consists of coding information into a 1-dimensional bit stream sequence. Then the encoded sequence is transmitted via transmission channels to decoder block , where the sequence of data is decoded. Decompression in the receiver realizes inverse operations i.e., channel decoder, and source decoder. The output image may or may not be an exact replica of original mage(lossless or lossy)[4].

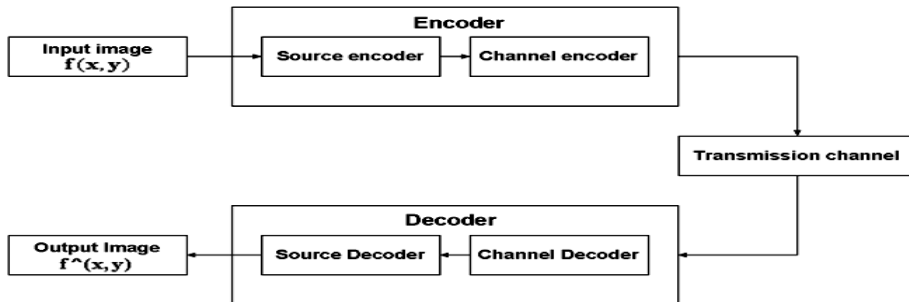


Figure 1: A general compression system model.

Before handling the proposed image compression scheme let us review briefly image compression techniques.

2.1. Image Compression Techniques

In general image compression methods are classified into two classes :lossless methods such as (Run length encoding - Huffman coding - Delta coding - Dictionary methods, etc),and lossy methods such as (Quantization-BTC-AMBTC-Transform coding , etc). In this section we will focus on BTC, AMBTC, Delta encoding and Huffman coding as follows:

2.1.1. BTC algorithm

The BTC algorithm involves the following steps in the coding phase[12]:

1. The input image is divided into M*N (typically 4 * 4) of non-overlapping blocks of pixels.
2. For each block the statistical moments: mean \bar{x} , and standard deviation σ are calculated using the following equations [12]:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

Where x_i represents the i^{th} pixel value of the image block, and n is the total number of pixels in that block. The two values \bar{x} and σ are termed as quantizes of BTC.

3. A two-level bit plane is obtained by comparing each pixel value x_i with the threshold value (\bar{x}). If $x_i < \bar{x}$ then the pixel is represented by '0', otherwise by '1'. The compressed data contains the bit plane along with \bar{x} and σ .

In the decoding phase, an image block is reconstructed by replacing '1' s in the bit plane with H and the '0' s with L, which are given by [12]:

$$H = \bar{x} + \sigma \sqrt{\frac{p}{q}} \quad (3)$$

$$L = \bar{x} - \sigma \sqrt{\frac{q}{p}} \quad (4)$$

Where p and q are the number of 0's and 1's in the compressed bit plane respectively.

2.1.2. AMBTC algorithm

The AMBTC algorithm involves the following steps in the coding phase [10]:

1. The input image is divided into M*N non-overlapping blocks (typically 4 * 4).The average gray level of the block is calculated using the following equations [10]:

$$\bar{x} = \frac{1}{16} \sum_{i=1}^{16} X_i \quad (5)$$

Where xi represents the ith pixels in the block.

2. The Pixels in the image block are classified into two ranges of values. The upper range is the gray levels which are greater than the block average gray level \bar{x} , and the lower range is the gray levels which are smaller than the block average gray level \bar{x} .
3. The mean of higher range X_H and the lower range X_L are calculated as [10]:

$$X_H = \frac{1}{k} \sum_{x_i \geq \bar{x}}^{16} X_i \quad (6)$$

$$X_L = \frac{1}{16 - k} \sum_{x_i < \bar{x}}^{16} X_i \quad (7)$$

Where k is the number of pixels whose gray level is greater than \bar{x} .

4. A two-level bit plane is obtained by comparing each pixel value xi with the threshold value (\bar{x}). If $x_i < \bar{x}$ then the pixel is represented by '0', otherwise by '1'.The compressed data contains the binary block along with X_H and X_L .

After generation of X_H , X_L , and bit plane, each block needs 32 bits (8 for X_H , 8 X_L , 16 for bit plane) to specify the bock data. Whereas the original block requires 128 bits (4*4*8).So that the compression ratio = 128/32 = 4.

On the other hand in the decoding phase, an image block is reconstructed by replacing the '1' s with X_H and the '0' s by X_L .The coding and decoding processes of AMBTC are faster than original BTC, because square root and multiplication operations are omitted [6].

2.1.3. Delta encoding

Delta encoding is a lossless data compression method. The principle is to represent data as the difference between successive samples rather than original samples [14]. Because delta encoding increases samples probability, and lowers the amplitude of signal. It's a common strategy for compression signals particularly, when followed by huffman or run-length encoding [14]. Table 1 shows an example of image compression based on delta encoding. The first value in the encoded stream is the same as the first pixel value in the original stream. Thereafter, each value in the encoded stream represents the difference between the current and last pixel value in the original stream.

Original Pixel	20	19	16	13	14	15	18	18	14	14	14	15	19	18	20
Delta Encoded	20	-1	-3	-3	1	1	3	0	-4	0	0	1	4	1-	2
Reconstructed Pixel	20	19	16	13	14	15	18	18	14	14	14	15	19	18	20

Table 1: Example of Delta Encoding.

It is noted in the previous example, the number of elements in the encoded stream are equal to the number of elements in the original stream. So delta encoding doesn't compress data alone but used as a pre-compression step to reduce interpixel redundancy from the image.

2.1.4. Huffman coding

Huffman coding is a commonly used method for lossless data compression. Since its development, in 1952, by D. Huffman [18]. Huffman coding is an entropy encoding algorithm used for lossless data compression. The principle is to use a lower number of bits to encode the data that occur more frequently. Codes are stored in a code table which may be constructed for each image or a set of images. In all cases the code table and encoded data must be transmitted to enable decoding [20].

Huffman method is based on the three conditions. The conditions are [21]:

1. The codes corresponding to the higher probability symbols could not be longer than the code words associated with the lower probability symbols.
2. The two lowest probability symbols had to have code words of the same length.

- The two lowest probability symbols have codes that are identical except for the last bit.

Example: Supposing an image with five pixels [P1, P2, P3, P4, P5], with probabilities of occurrence $P(P1) = 0.15$, $P(P2) = 0.04$, $P(P3) = 0.26$, $P(P4) = 0.05$, and $P(P5) = 0.50$. Huffman encoder uses the following steps for generating variable size codes:

- Calculating the entropy rate as follows:

$$H = -\sum_{i=1}^5 P(P_i) \log_2 P(P_i) = 1.817684 \text{ bits} \quad (8)$$

Where $P(P_i)$ is the occurrence probability of each P_i .

- Sorting the image pixels in descending order of their probabilities as shown in the following table:

Pixel	Probability
P1	0.50
P2	0.26
P3	0.15
P4	0.05
P5	0.04

Table 2

- Building the binary tree as follows:

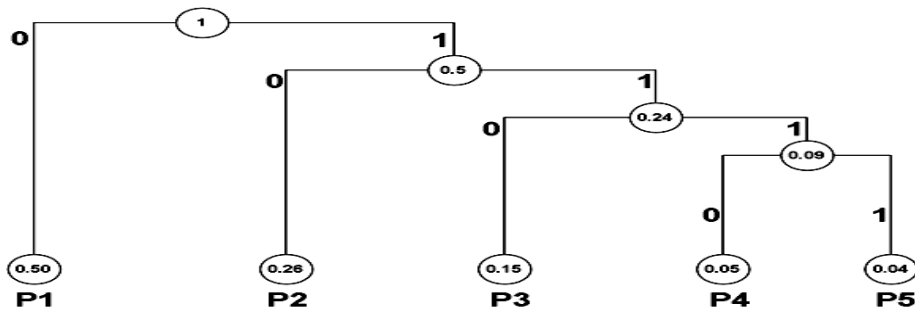


Fig 2: Huffman code for the five pixel.

- Generating a Huffman code dictionary for each pixel, as shown in table 3.

Pixel	code
P1	0
P2	10
P3	110
P4	1110
P5	1111

Table 3.

5. Replacing each pixel in the original image with the respective code in the dictionary. Where the pixels more frequent are coded with smaller number of bits.
6. Calculating the average length of the number of bits used to represent each pixel , which is defined as [4]:

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

Where $l(r_k)$ is the length of the codeword used in pixel r_k , and $Pr(r_k)$ is the occurrence probability of each r_k .

$L_{avg} = (0.5)*(1) + (0.26)*(2) + (0.15)*(3) + (0.05)*(4) + (0.04)*(4) = 1.83$ bits/pixel. But the entropy of the source pixels is 1.817684 bits/pixel. So the resulting huffman code efficiency is $1.817684 / 1.83 = 0.9933$.

The major advantages of huffman coding are easy to implement, produce a lossless compression of images [20]. So it is widely used in many applications such as JPEG , DEFLATE [20], and compression softwares like pkZIP, lha, gz, zoo and arj [22]. On contrast , the main drawbacks of huffman coding are a relatively slow process , its efficacy depends on the accuracy of the statistical model used and type of image , decoding is difficult due to different code lengths , causing overhead due to code table must be transmitted at the beginning of the compressed file [20], using an integral number of bits in each code, and not producing very good compression ratios [23].

As illustrated previously, AMBTC produce a fixed length binary representation of each block according to the block size of the image , for example (compression ratio = 4; block size = 4*4). On the contrary huffman coding produce a variable length binary representation , based on the

statistical nature of the image. So our scheme proposed to improve the performance of AMBTC to produce a variable length compression by applying the combination of delta encoding, and Huffman coding. In the following section we will introduce the proposed scheme.

3. Proposed scheme

In a color image high correlation exists among R, G, and B planes, so a high compression ratio can be achieved by exploiting the psychovisual, spatial correlations, and the coding redundancies. In the proposed method the psychovisual redundancy is reduced by converting RGB to a less correlated color space such as YCbCr. On the other hand the spatial redundancy is reduced by block quantization using AMBTC method, and delta encoding. Finally the coding redundancy is reduced by Huffman coding.

The procedures of proposed scheme for color image compression are shown in figure 3. In the pre-processing step, color image is transformed from RGB color space into another less correlation color space such as luminance/chrominance to generate Y, Cb, Cr components. Since the human eye is more sensitive to luminance changes than chrominance variance, the chrominance components are downsampled to reduce the size of the original image. In step 1, the three components (Y, Cb, Cr) go through AMBTC encoder independently to reduce the spacial redundancy. In the second step, delta encoding is used to increase the appearance of each pixel in the image. Finally in step 3, Huffman coding is applied to achieve compression.

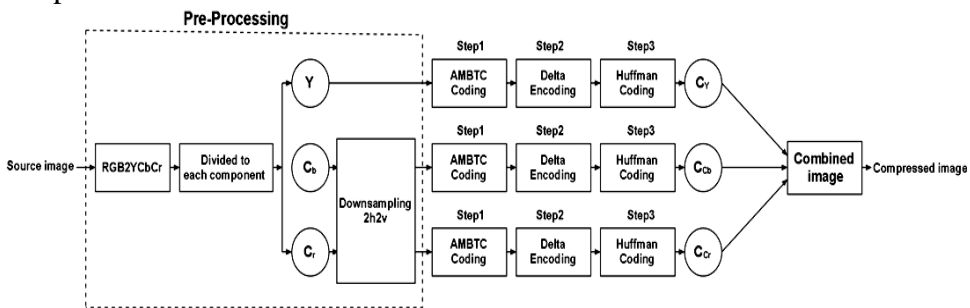


Figure 3. Compression steps using the proposed method.

The details of the encoding and decoding phases of the proposed algorithm will be illustrated in the next section.

3.1. Encoding Phase

1. Convert the color image from RGB color space to YC_bC_r color space for better coding efficiency. Since the human eye is more sensitive to small changes in luminance but not in chrominance, so the chrominance part can lose much data, without affect on image quality.
2. Divide the converted image is into three matrices: Y, C_b, C_r .
3. Downsample the chroma matrices (C_b, C_r) at a ratio of 2:1 both horizontally and vertically (is called 2h2v).
4. Compress each matrix of the three matrices (Y, C_b, C_r), independently, as follows :
 - 4.1. Divide the given image into a set of non overlapping blocks, the size of a block could be $4*4$ or $8*8$.
 - 4.2. Apply AMBTC principles for each block, as follows:
 - 4.2.1. Calculate the average gray level of the block (\bar{x}).
 - 4.2.2. Compute the lower mean X_L , and higher mean X_H of the block.
 - 4.2.3. Use the binary block to represent the pixels, where "1" is used to represent a pixel whose gray level is grater than or equal to (\bar{x}), and "0" is used to represent a pixel whose gray level is less than (\bar{x}).
 - 4.3. Construct the three vectors (BitmapVec, HighVec, LowVec). Where BitmapVec involves all binary values in all blocks of the image. High includes all X_H values in the image, and LowVec contains all X_L values in the image
 - 4.4. Convert the BitmapVec to DecVec (from binary representation to deciaml system).
 - 4.5. Calculate the difference between HighVec, LowVec, to get the difference vector (DiffVec). Which contains small close values.
 - 4.6. Apply delta encoding for each vector of the three vectors (DecVec, LowVec, DiffVec), to get the delta vectores (DeltaDec, DeltaLow, DeltaVec).
 - 4.7. Apply huffman coding for each vector of the three vectors (DeltaDec, DeltaLow, DeltaDiff), independently to generate compressed files (Comp1, Comp2, Comp3).
 - 4.8. Combine the encoded data and side information for compreseed files into a single component.

5. Finally, combine the compressed components (C_Y, C_{Cb}, C_{Cr}), which are generated from previous steps to generate the compressed file, before storage or transmission process.

3.2. Decoding Phase

The procedures of the decoding phase is reverse to that of the encoding phase as shown in fig4. The details of the decoder algorithm are described as follows:

1. Apply the following steps for each component of the compressed file (C_Y, C_{Cb}, C_{Cr}) independently:
 - 1.1. Apply huffman decoding into compressed files (Comp1-Comp2-Comp3), to restore original vectors (DeltaDec, DeltaLow, DeltaDiff).
 - 1.2. Apply delta decoding into vectors (DeltaDec, DeltaLow, DeltaDiff), to get the three vectors (DecVec, LowVec, DiffVec).
 - 1.3. Construct the high values vector (HighVec).
 - 1.4. Convert the decimal numbers vector (DecVec) to binary matrix (BitmapMat).
 - 1.5. Apply AMBTC decoding principles into BitmapMat. Where it is divided into $M \times N$ non-overlapping blocks. For each block, 0's is replaced by the corresponding number in a LowVec, and 1's is replaced by the corresponding number in a HighVec.
2. Upsample the chroma matrices (R_{Cb}, R_{Cr}), both horizontally and vertically, to get the reconstructed chroma matrices with original image dimensions (R_{Cb}, R_{Cr}).
3. Combine the three reconstructed matrices (R_Y, R_{Cb}, R_{Cr}) into a single matrix.
4. Convert the reconstructed matrix from $YCbCr$ color space to RGB color space.

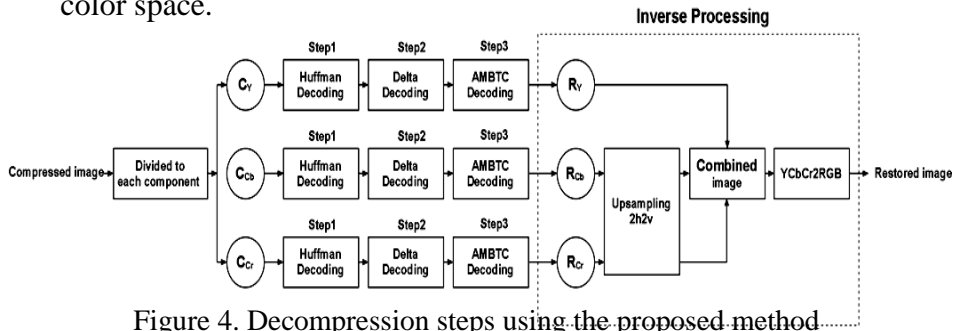


Figure 4. Decompression steps using the proposed method.....

4. Experimental Results

To evaluate the performance of the proposed image compression scheme, we took seven standard color images of size 512*512 (24 bit per pixel) namely “Lena” , ”Peppers” , “Mandrill” , “Girl” , ”Airplane” , ”House“ , and “Sailboat “ which are shown in Fig 5. The Measurement criteria required to assess the performance of proposed method are: the compression ratio (CR) given by Eq. (10) , the root mean square error (RMSE) given by Eq. (11) , and the peak signal to noise ratio (PSNR) given by Eq. (12) [13 , 24 , 25].

CR = size of original image /size of compressed image

$$RMSE = \left[\frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y [I(x,y) - I'(x,y)] \right]^{1/2}$$

Where $I(x,y)$ is the original image , $I'(x,y)$ is the reconstructed image, and $X*Y$ is the dimensions of the images.

$$PSNR = 20 \log \left(\frac{255}{RMSE} \right)$$

Where typical PSNR values range between 20 and 40 dB.

A smaller value of the RMSE means that the reconstructed image has less distortion. In contrast a higher value of PSNR means lesser error in the reconstructed image. So a compression method having lower RMSE and correspondingly high PSNR values could be recognized as a better scheme.



Figure 5: Standard color images used for experiment (*).

* <http://sipi.usc.edu/database/index.php?volume=misc&image=11#top>.

Experimental results using the proposed scheme on the taken standard color images are compared with the original BTC and AMBTC. The obtained results for block size of 4*4 are given in table 4, and presented in fig 6 and fig 7. While the results for block size of 8*8 are given in table 5, and presented in fig 8 and fig 9.

Image Name	BTC			AMBTC			Proposed Method		
	Cr	RMSE	PSNR	Cr	RMSE	PSNR	Cr	RMSE	PSNR
Lena	4	5.8437	32.797	4	5.6166	33.1413	12.84	8.1674	29.8891
Peppers	4	6.1312	32.3799	4	5.8888	32.7303	12.39	11.9579	26.5777
Mandrill	4	12.7458	26.0234	4	12.232	26.3808	11.87	18.2294	22.9153
Girl	4	4.515	35.0377	4	4.3405	35.3799	13.11	9.9332	28.189
Airplane	4	6.3151	32.1232	4	6.0871	32.4426	13.17	9.5377	28.5419
House	4	7.8992	30.1792	4	7.5974	30.5175	12.90	11.9102	26.6124
Sailboat	4	9.1172	28.9336	4	8.7512	29.2895	12.21	14.0281	25.1908
Average	4	7.510	31.068	4	7.216	31.412	12.641	11.966	26.845

Table 4: Experimental results of BTC, AMBTC, proposed method at block size 4*4.

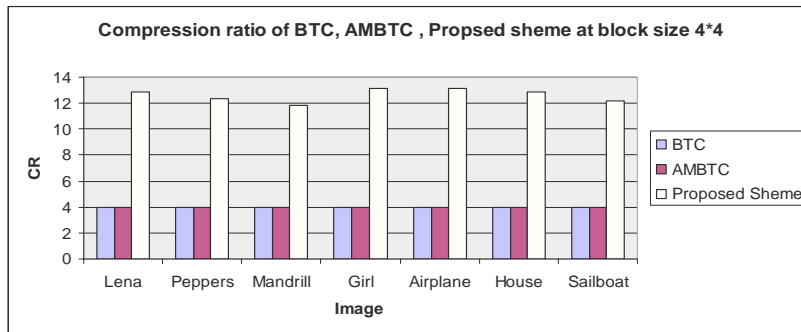


Figure 6.

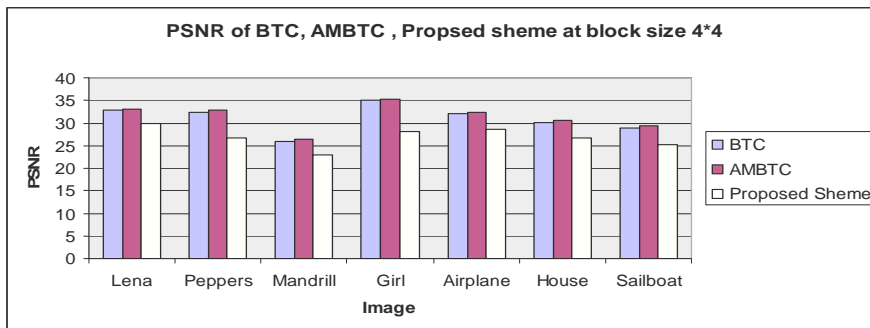


Figure 7.

Image Name	BTC			AMBTC			Proposed Method		
	Cr	RMSE	PSNR	Cr	RMSE	PSNR	Cr	RMSE	PSNR
Lena	6.4	8.3603	29.6864	6.4	8.0179	30.0496	19.58	10.3263	27.8519
Peppers	6.4	9.3082	28.7535	6.4	8.9106	29.1327	18.27	14.3768	24.9776
Mandrill	6.4	15.802	24.1566	6.4	15.1202	24.5396	17.72	20.5009	21.8953
Girl	6.4	6.4779	31.9021	6.4	6.1956	32.2892	19.36	10.9281	27.3599
Airplane	6.4	9.1272	28.924	6.4	8.7842	29.2567	19.94	11.6978	26.7687
House	6.4	11.4553	26.9507	6.4	10.9795	27.3192	20.06	14.4308	24.945
Sailboat	6.4	12.7619	26.0125	6.4	12.2221	26.3879	18.29	16.5912	23.7332
Average	6.4	10.470	28.055	6.4	10.033	28.424	19.031	14.121	25.361

Table 5: Experimental results of BTC, AMBTC, proposed method at block size 8*8.

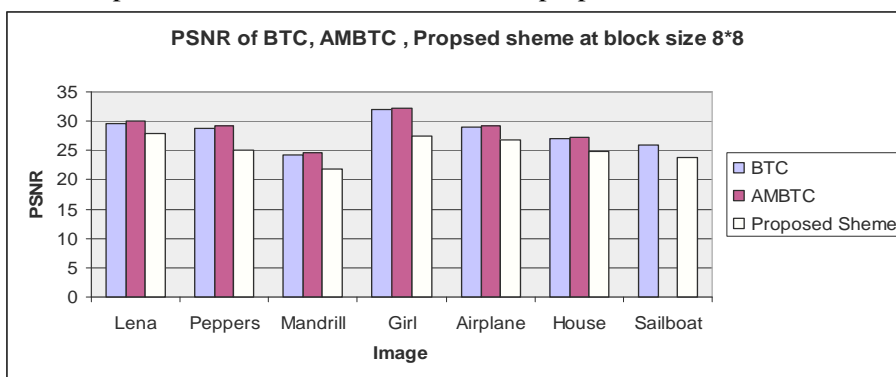


Figure 8.

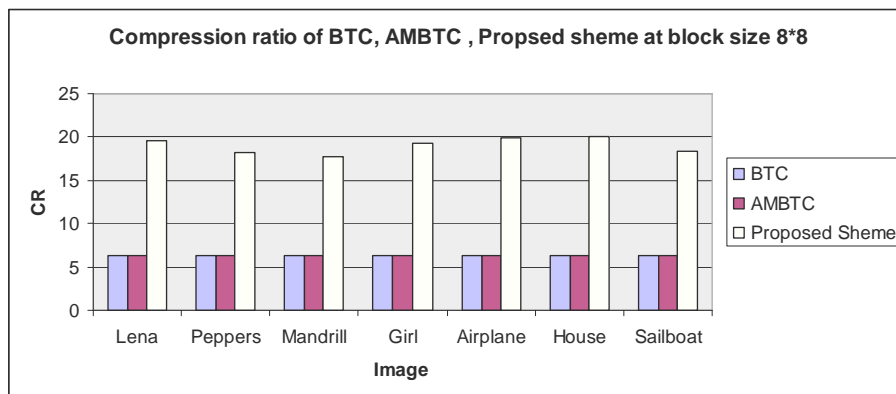


Figure 9.

The above tables assure that the image compression using AMBTC provides better image quality than image compression using BTC at the same compression ratio. While , the proposed scheme achieves higher compression ratio than that of AMBTC scheme with small degradation of image quality. Since the human eye is more sensitive to small changes in luminance but not in chrominance, so the chrominance part can lose much data, without introducing noticeable degradation in the reconstructed images.

5. Conclusion

A modified method for improving the conventional AMBTC has been proposed. The proposed method uses delta encoding to increase the appearance degree of image pixels , and huffman coding to generate variable length compression. The performance of the proposed method has been compared with conventional BTC and AMBTC and it is found that it achieves a higher compression ratio than that of both BTC , and AMBTC with low distortion of decoded images. Experimental results, by applying our scheme on standard seven color images, show that an average compression ratio for block size 4*4 is 12.641 while for block size 8*8 is 19.031. On the other hand an average PSNR value for block size 4*4 is 26.845 , and for block size 8*8 is 25.361. Our compression scheme may be useful for low cost handheld devices with low computational power to handle images.

References

- [1] Munaga V.N.K. Prasad , V.N. Mishra , K.K. Shukla, Space Partitioning Based Image Compression Using Quality Measures For Subdivision Decision, Applied Soft Computing Journal, 3 (3) (2003) 273-282.
- [2] C. Saravanan , R. Ponalagusamy, Lossless Grey scale Image Compression using Source Symbols Reduction and Huffman Coding, international Journal of Image Processing (IJIP), 3 (5) (2009) 246-251.
- [3] JAGADISH H. PUJAR & LOHIT M. KADLASKAR, A New Lossless Method Of Image Compression And Decompression Using Huffman Coding Techniques , Journal Of Theoretical And Applied Information Technology , 15 (1) (2010) 18-22.
- [4] Vo Si Van, Image Compression Using Burrows-Wheeler Transform, Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology, HELSINKI UNIVERSITY OF TECHNOLOGY , (2009).

- [5] E.J.Delp, O.R.Mitchell, Image Coding Using Block Truncation Coding. IEEE Transactions on Communications, 27 (1979) 1335-1342.
- [6] P.Franti, O.Nevalainen, T.Kaukoranta, Compression of Digital Images by Block Truncation Coding: A Survey, The Computer Journal, 37 (4) (1994) 308-324.
- [7] Søren I. Olsen, Block truncation and planar image coding, Pattern Recognition Letters 21 (2000) 1141-1148
- [8] Chung-Woei Chao, Chaur-Heh Hsiehb, Po-Ching Lu ,Taj-An Cheng , modified block truncation coding for image compression , Pattern Recognition Letters , 17 (1996) 1499-1506.
- [9] Bibhas Chandra Dharaa, Bhabatosh Chandab, Block truncation coding using pattern fitting , Pattern Recognition , Pattern Recognition 37 (2004) 2131 – 2139
- [10] K.Somasundaram and I.Kaspar Raj, Low Computational Image Compression Scheme based on Absolute Moment Block Truncation Coding, World Academy of Science, Engineering and Technology , 19 (2006) 166-171 .
- [11] Edward J. Delp, Martha Saenz, and Paul Salama, BLOCK TRUNCATION CODING (BTC), Handbook of Image and Video Processing, Processing, edited by Bovik A. C., Academic Press, (2000) 176-181.
- [12] K.Somasundaram and I.Kaspar Raj, An Image compression Scheme based on Predictive and Interpolative Absolute Moment Block Truncation Coding, GVIP Journal , 6 (4) (2006) 33-37.
- [13] 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) , (2011) 9-13.
- [14] Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing , 2rd, California Technical Publishing, (1999).
- [15] Torsten Seemann, Digital Image Processing using Local Segmentation, Submission for the degree of Doctor of Philosophy, (2002).
- [16] Matlap Helper.
- [17] Nageswara Rao Thota, Srinivasa Kumar Devireddy, Image Compression Using Discrete Cosine Transform, Georgian Electronic Scientific Journal: Computer Science and Telecommunications , 3 (17) (2008) 35-43.
- [18] David Salomon, Data Compression The Complete Reference, 3rd, Morgan Kaufmann Publishers, (2004).
- [19] Yun Q. Shi and Huifang Sun, Image and video compression for multimedia engineering : fundamentals algorithms and standards, 2rd , CRC Press , (2008)
- [20] Mamta Sharma, Compression Using Huffman Coding, JCSNS International Journal of Computer Science And Network Security, 10 (5) (2010) 133-141.
- [21] Khalid Sayood, Data Compression , available on line on www.sciencedirect.com.

- [22] http://www.prepressure.com/library/compression_algorithms/huffman.
- [23] Mark Nelson & Jean loup Gailly , The Data Compression Book , M&T Books, New York, 2rd , (1995).
- [24] T.M. Amarunnishad, V.K. Govindan , Abraham T. Mathew, Improving BTC image compression using a fuzzy complement edge operator, Signal Processing 88 (2008) 2989-2997.
- [25] Amhamed Saffor, Abdul Rahman Ramli, A COMPARATIVE STUDY OF IMAGE COMPRESSION BETWEEN JPEG AND WAVELET, Malaysian Journal of Computer Science, 14 (1) (2001).