INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & MANAGEMENT

PERFORMANCE ANALYSIS OF BTC-PF ALGORITHM FOR IMAGE COMPRESSION

Priya Nagar[1], Prof. A.C.Tiwari[2], Prof. Mukesh Patidar[3]
[#1]MTech Scholor, Department of Electronics & Communication, LNCT, Indore
[#2] Head of Department, Department of Electronics & Communication, LNCT, Indore
[#3]Assistant Professor, Department of Electronics & Communication, LNCT, Indore
priya19a@yahoo.co.in

ABSTRACT

In this paper, a novel Gray Scale image compression method (BTC algorithm using Pattern fitting) based on block truncation coding (BTC) and Vector quantization (VQ) is proposed. This method inherits the advantages of both BTC and vector quantization. The BTC-PF method has some controlling parameters through which we can control the bit-rate and quality. The BTC-PF method is also successfully employed to handle low bit-rate coding.

The performance of proposed image compression method is quite good. As the decoding methods are mainly based on table look up, decoding module of these methods are faster than that of the state-of-the-art techniques. The effectiveness of the proposed schemes is established by comparing the performance with that of the existing methods.

***Keywords*— Image Processing, Compression, BTC Scheme, storage, resolution, decoding, transmission, MSE, PSNR, SSIM.**

I. BLOCK TRUNCATION CODING

Block truncation coding (BTC) is a simple, fast, lossy and fixed length compression technique for gray scale images. This is a block-adaptive binary encoding scheme based on moment preserving quantization. The concept is introduced by Delp and Mitchell [2,8,38]. In BTC method, an image is divided into n×n blocks (in general n=4) and each block is coded separately. Gray levels of each block is quantized by Q level quantizer and these quantizer levels are chosen such that a few low order moments are preserved in the quantized output. In the simplest form of BTC, the first two moments are preserved and blocks are represented by two quantization levels. By incorporating additional constraints, higher order moments can be preserved. Suppose k ($=n^2$) be the number of pixels in a block and also suppose f(xi), xi ∈ C are the gray values of the pixels in a block of the original image where C represents the set of coordinates of pixels in the block, i.e., $C = \{x_1, x_2, \dots \dots . x_k\}$.
The first two sample moments m_1 and m_2 are given by

$$m_1 = \frac{1}{k}\sum_{i=1}^{k} f(x_i)$$

$$m_2 = \frac{1}{k}\sum_{i=1}^{k} f(x_i)^2$$

m_1 is the sample mean and the sample variance σ^2 of image block is given by

$$\sigma^2 = m_2 - {m_1}^2$$

Then a two level quantization is performed on the block. The pixels with intensity greater than the quantization threshold are quantized to value b, and the other pixels are quantized to value a. Here, the sample mean m_1 is set as the quantization threshold and suppose the quantization partition of C with respect to the threshold m_1 gives two sets of pixels C_0 and C_1, such that [10,11,12],

C = $C_0 \cup C_1$ and $C_0 \cap C_1 = \emptyset$ where,

$$C_0 = \{x'_1, x'_2, \dots \dots \dots , x'_{k'}\} \text{ and } C_0 = \{x''_1, x''_2, \dots \dots \dots , x''_{k-k'}\}$$

The quantization partition of C can be represented by a binary-pattern P. The binary pattern or bit-plane P is defined as

$$P(x_i) = \begin{cases} 0 \; if \; f(x_i) \leq m_1 \\ 1 \; if \; f(x_i) > m_1 \end{cases}$$

The partition of C is defined as

$$C_0 = \{x_i | p(x_i) = 0\} \; and \; C_1 = \{x_i | p(x_i) = 1\}$$

The quantization levels a and b are used to preserve first two sample moments. Then

$$km_1 = k'a + (k - k')b$$
$$km_2 = k'^{a^2} + (k = k')b^2$$

And solving for a and b yields

$$a = m_1 - \sigma \sqrt{\frac{k - k'}{k'}}$$

$$b = m_1 + \sigma \sqrt{\frac{k'}{k - k'}}$$

Thus a compressed image block is represented by the triplet (a, b, P). For 8-bit gray scale image, the straightforward BTC needs $\frac{8+8+16}{16} = 2$ bits per pixel. An example of BTC method is shown in Fig. 1.

2	9	12	15
2	11	11	9
2	3	12	15
3	3	4	14

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

2	12	12	12
2	12	12	12
2	2	2	12
2	2	2	12

Figure 1: An example of BTC: (a) original block, m1 = 7.94, σ = 4.91, (b) corresponding bit-pattern, k0 =7, (c) reconstructed block with a=2 and b=12.

The steps for encoding a block by BTC are:

- Perform quantization (determination of threshold, and quantization levels)
- Coding of the quantization levels (i.e., a and b)
- Coding of the bit-pattern P

BTC method is simple and results into high quality reconstructed image, but the compression ratio is low. In literature, there are several modifications of BTC to improve the performance [1,2,5].

The original BTC uses threshold $T_h = m_1$ and preserves the first and second order moments. This method can be extended to preserve third order moments also. To preserve third order moment, the T_h is determined such a way that number of pixels greater than T_h is given by:

$$q = (k - k') = \frac{k}{2}\left(1 + G\sqrt{\frac{1}{G^2 + 4}}\right)$$

Where

$$m_3 = \frac{1}{k}\sum_{i=1}^{k} f^3(x_i)$$

$$G = \frac{3m_1m_2 - m_3 - 2m_1^3}{\sigma^3}, \qquad \sigma \neq 0$$

The absolute moment BTC (AMBTC) [39] preserves sample mean (m1) and first order absolute central moment α, where

$$\alpha = \frac{1}{k}\sum_{i=1}^{k} |f(x_i) - m_1|$$

and corresponding quantization levels are

$$a = m_1 - \frac{\alpha k}{2k'}$$

$$b = m_1 + \frac{\alpha k}{2(k - k')}$$

If the quantization levels are set as m_{low} (lower mean) and m_{high} (higher mean), then MSE is reduced. The means, m_{low} and m_{high} are defined as follows.

$$m_{low} = \frac{1}{k'}\sum_{f(x_i \le m_1)} f(x_i)$$

$$m_{high} = \frac{1}{k - k'}\sum_{f(x_i > m_1)} f(x_i)$$

AMBTC is faster than BTC, as no square root operation is required.

In the bit-stream of BTC method, bit-plane shares a significant amount of space. Such requirements can be reduced. Normally, lossless coding for bit-plane is acceptable. However, considering a lossy coding method for bit-plane a large compression is possible at the cost of the quality [13,14,15].

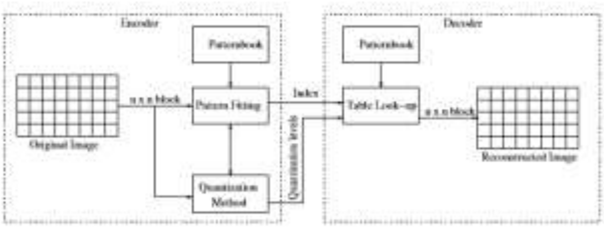

Figure 2: A schematic diagram of BTC-PF method.

II. BTC-PF METHOD

The encoding method of VQ is time consuming, whereas its decoding method uses table look-up method and is very fast. This method results in higher compression ratio, though quality of the reconstructed image is usually not as good as BTC. BTC is a simple and fast method, which enables high quality reconstruction but bit-rate is also high. Comparatively, the encoder of BTC is faster than that of VQ, while its decoder is little slower. A compromise between these two methods gives a fast decoder, maintains good quality for reconstructed image with moderate bit-rate. Again, this hybrid method can also be used in image feature extraction. That means the compressed data due to this method can directly be used to compute image features like, edge histogram and so on. In BTC, a block is represented by (a,b,P), where a and b are quantization levels for reconstruction and P represents the intensity pattern within the block. The pattern P shares a significant amount of space in the bit-

stream of the BTC method. In VQ based BTC method there is a binary codebook and instead of transmitting actual bit-pattern, the index of the closest bit-pattern (from the codebook) is transmitted [16,17,19].
In the proposed BTC-PF method, for each image block of size n×n, the pattern P is selected from a set of, say, M predefined patterns called pattern book. Each pattern represents Q intensity levels of the block and size is same as that of the block (i.e., n × n). For an image block B, the best fit pattern P is found from the pattern book. Then Q quantization levels are determined from the B using the intensity pattern defined by the selected pattern P. At the time of reconstruction of a block, the index of the selected pattern and Q different gray levels are required. As the method selects a pattern from a set of predefined patterns to represent the block, the quality of the reconstructed image is, in general, little lower than that of conventional BTC. However, this little sacrifice in PSNR earns a huge gain in bit-rate. The block diagram of the BTC-PF method is shown in Fig.2. Like VQ methods, the performance of BTC-PF also depends on the pattern book. Hence, the design of pattern book is a crucial step [6,8,9].

The encoder of BTC-PF thus consists of:

(i) pattern fitting (selection of the best pattern) and
(ii) The quantization method, i.e., determining the Q gray levels.

Here also, the decoder uses the table look-up method.

III. IMPLEMENTATION OF BTC-PF ALGORITHM FOR GRAY SCALE IMAGE

Implementation of the proposed methodology has been carried out for different images and their performance has been evaluated for Gray Scale Image. Different block size has been chosen for this lossy compression technique, of 4×4 block size, 8×8 block size and 16×16 block size. PSNR and MSE have been evaluated for performance analysis.

III.I Implementation of BTC-PF Algorithm on BLOCK SIZE 4×4

For the block size of 4×4, first the image (m×n) is divided in a block of size 4×4 then if there is any residue it is adjusted to fit in this size. BTC –PF algorithm is applied on this image and the image is compressed. The obtained image having comparable resolution than the original image but the image size is reduced significantly. This algorithm is applied for different images and result has been analyzed. Five different Image of size 255×255 has been taken and converted to the Gray Scale Image, by applying the BTC-PF algorithm image has been compressed.

Figure 3: BTC-PF Algorithm applied on Lena Image of block size 4×4: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 4: BTC-PF Algorithm applied on Baboon Image of block size 4×4: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image.

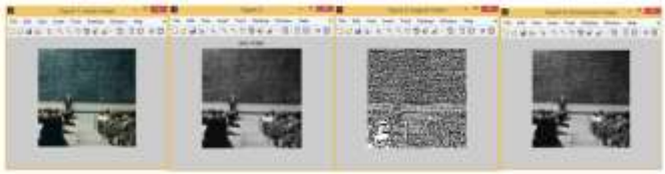

Figure 5: BTC-PF Algorithm applied on Class Image of block size 4×4: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 6: BTC-PF Algorithm applied on Bike Image of block size 4×4: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 7: BTC-PF Algorithm applied on Mall Image of block size 4×4: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Table 4.3: PSNR, MSE, CR, SSIM calculation for block size 4×4

Image of Size 256*256	MSE	PSNR	CR	SSIM
Lena Image	16.0643	81.8468	1.625	0.9965
Baboon Image	32.2887	72.9427	1.525	0.9952
Class Image	23.4409	78.2339	1.567	0.9922
Bike Image	12.5983	85.4110	1.57	0.9962
Real Image	16.8051	82.6084	1.625	0.9947
Average Value	20.2395	80.2085	1.582	.995

III.II Implementation of BTC-PF Algorithm on BLOCK SIZE 8×8

Same approach as discussed in Implementation of BTC-PF Algorithm for Block size 4×4, is adopted for block size 8×8. It is worth noting aspect to check the image dimension is fitting the block size or not. The PSNR & MSE is calculated based on the result. The result obtained in block size 8×8 is poor performance in terms of resolution but far better in terms of size.

It is suggested that when image quality is comprisable this compression technique provides good performance in terms of storage capability. It is suggested to keep the block size of image less because increasing the number of order of block size will result poorer resolution of the image but compression ratio will be far better if we are increasing the number of block size.

Figure 8: BTC-PF Algorithm applied on Lena Image of block size 8×8: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

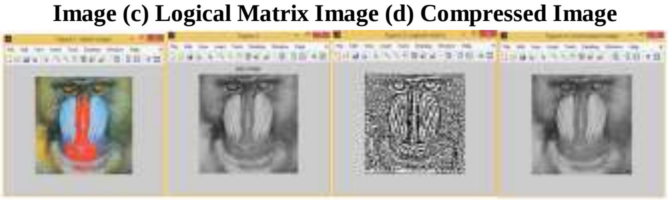

Figure 9: BTC-PF Algorithm applied on Baboon Image of block size 8×8: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 10: BTC-PF Algorithm applied on Class Image of block size 8×8: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 11: BTC-PF Algorithm applied on Bike Image of block size 8×8: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 12: BTC-PF Algorithm applied on Mall Image of block size 8×8: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Table 4.3: PSNR, CR, MSE & SSIM calculation for block size 4 x 4

Image of Size 256*256	MSE	CR	PSNR	SSIM
Lena Image	26.5617	2.6	76.8181	0.9933
Baboon Image	44.0471	2.25	69.8373	0.9925
Class Image	32.1356	2.41	75.0791	0.9861
Bike Image	19.8806	2.36	80.8493	0.9929
Real Image	28.2186	2.4	77.4255	0.9895
Average Value	30.1687	2.41	76.0018	0.9908

III.III Implementation of BTC-PF Algorithm on BLOCK SIZE 16×16

Methodology used to implement the BTC-PF Algorithm for block size 16×16 will be same as above discussed. This block size division criterion gives poor result than 4×4 o4 8×8 block size in terms of resolution but in terms of compression, compressed image size is smaller than 4×4 o4 8×8 block size.

Figure 13: BTC-PF Algorithm applied on Lena Image of block size 16×16: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 14: BTC-PF Algorithm applied on Baboon Image of block size 16×16: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 15: BTC-PF Algorithm applied on Class Image of block size 16×16: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 16: BTC-PF Algorithm applied on Bike Image of block size 16×16: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Figure 17: BTC-PF Algorithm applied on Mall Image of block size 16×16: (a) Original image of size 255*255 (b) Gray Image (c) Logical Matrix Image (d) Compressed Image

Table 4.3: PSNR, CR, MSE & SSIM calculation for block size 16×16

Image of Size 256*256	MSE	CR	PSNR	SSIM
Lena Image	38.4451	3.71	73.1205	0.9897
Baboon Image	53.3275	3.05	67.9254	0.9901
Class Image	38.7285	3.22	73.2130	0.9793
Bike Image	29.3751	3.46	76.9452	0.9883
Real Image	37.8205	3.46	74.4968	0.9847
Average Value	39.5394	3.38	73.1402	.9864

IV. RESULT & DISCUSSION

In the present era of multimedia, the requirement of image storage and transmission for image are increasing exponentially. As a result, the need for better compression technology is always in demand. The limited bandwidth of internet also asks for transmission of desired objects only. Progressive image transmission provides such facilities, where transmission is done in steps and the transmission of undesired image can be stopped at an early stage. Modern applications, in addition to high compression ratio, also demand for efficient encoding and decoding processes, so that computational constraint of many real-time applications is satisfied. In some

applications, multimedia objects are compressed just once but are reconstructed frequently and most of the times the compressed objects are available on the local storage device. In such cases, the compression methods should be such that efficient reconstruction of the objects must be possible along with high quality.
This paper concentrates on image coding that leads to fast decoding. Such techniques are suitable for image search and retrieval.

IV.I Implementation of BTC-PF Algorithm on Gray Scale Image

Implementation of the proposed methodology has been carried out for different images and their performance has been evaluated for Gray Scale Image. Different block size has been chosen for this lossy compression technique, of 4×4 block size, 8×8 block size and 16×16 block size. PSNR and MSE have been evaluated for performance analysis.

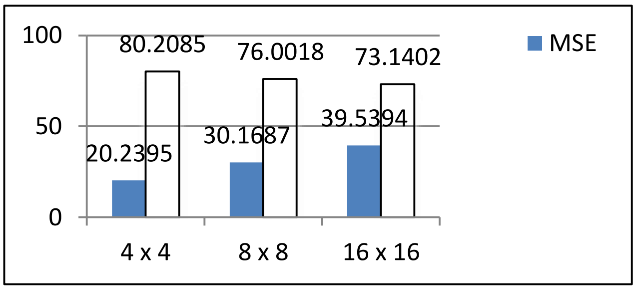

Figure 18 Comparison Plot of MSE & PSNR for Gray Scale Image Different Block Sizes

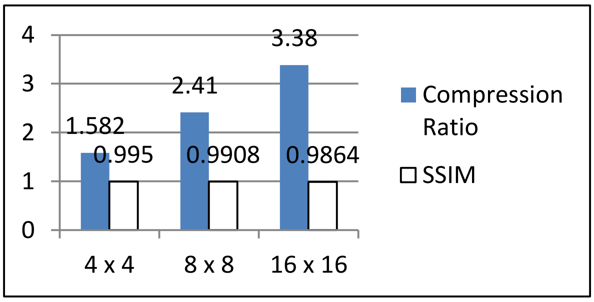

Figure 19: Comparison Plot of CR & SSIM for Gray Scale Image of Different Block Sizes

V. CONCLUSION

In this paper a spatial domain technique for image data compression, namely, the block truncation coding (BTC) has been considered. This technique is based on dividing the image into (4×4) non overlapping blocks and uses a two-level quantize. An advance form of BTC Algorithm has been proposed, termed as BTC-PF algorithm. The BTC-PF algorithm technique has been applied to Gray Scale image (Reference Image as well as Real Image) each contains 255×255 pixels. BTC-PF includes BTC algorithm as well as vector quantization method for the purpose of pattern fitting.
The reconstructed images obtained from applying this technique have excellent performance. For a block size of 4*4, MSE for real image is least, PSNR value is highest, CR is least and SSIM is highest. The same scenario is with an image of block size 8*8 & 16*16 respectively. But as we increase the block size performance of the algorithm degraded, i.e. we get blurred image. But memory space need to store the image is very less. So if user can compromise with the quality 16 *16 block size takes least memory space. But if balance between the memory size and image quality is needed block size of 8*8 is the best option.
Note: This algorithm corresponds to 75% compression. The peak signal-to-noise ratio is used as a measure of the reconstructed image quality comparison of the original and reconstructed image shows that this method provides a good compression without seriously degrading the reconstructed image.

REFERENCE

[1] D. Salomon, Data Compression: The Complete Reference. Springer-Verlag, New York, 2000.
[2] Y. Q. Shi and H. Sun, Image and video compression for multimedia engineering: fundamentals, algorithms, and standards. CRC Press, USA, 1999.
[3] D. Huffman, A method for the construction of minimum redundancy codes. Proceeding of the IRE, 40:1098 – 1101, 1952.
[4] I. H. Witten, M. N. Radford, and J. G. Cleary, Arithmetic coding for data compression. Communications of ACM, 60:520 – 540, 1987.
[5] W. Pratt, Digital Image Processing, John Wiley and Sons, New York, 1978.
[6] K. R. Rao and P. C. Yip Eds, The Transform and Data Compression Handbook. CRC Press, Boca Raton, FL, 2001.
[7] S. Dunn. Digital color. http://davis.wpi.edu/~matt/courses/color.
[8] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall , 2002.
[9] W. B. Pennebaker and J. L. Mitchell, JPEG: still image data compression standard. New York: Van Nostrand Reinhold, 1993.
[10] G. K. Wallace, The jpeg still picture compression standard. IEEE Transactions on Consumer Electronics, 38: xviii – xxxiv, 1992.
[11] D. Tabuman and M. W. Marcelin, JPEG 2000: image compression fundamentals, standards and practices, Kluwer Academic Publishers, 2001.
[12] A. Christopoulos, A. Skodras, and T. Ebrahimi, Jpeg2000 still image coding system: an overview. IEEE Transactions on Consumer Electronics, 46:1103–1127, 2000.
[13] A. Skodras, A. Christopoulos, and T. Ebrahimi, Jpeg2000: the upcoming still image compression standard. Pattern Recognition Letters, 22:1337–1345, 2001.
[14] T. Ebrahimi, C. A. Christopoulos, and D. T. Lee, Special issue introduction, jpeg2000. Signal Processing: Image Communication, 17:1–144, 2002.
[15] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients. IEEE Transactions on Signal Processing, 41:3445 – 3462, 1993.
[16] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on Circuits and System for Video Techniques, 6:243– 250, 1996.
[17] D. Marpe and H. L. Cycon, Very low bit rate video coding using wavelet-based techniques, IEEE Transactions on Circuits and Systems for Video Technology, 9:85 – 94, 1999.
[18] D. Gleich, P. Planinsic, and Z. Cucej, Low bitrate video coding using wavelet transform, In Proc. of the EURASIP conference on Video/Image Processing and Multimedia Communications, pages 369 – 374, 2003.
[19] R. Leonardi and J. R. Ohm, Wavelet video coding - an overview (iso/iec jtc1/sc29/wg11 w7824). 2006.
[20] D. Salomon. Data Compression: The Complete Reference. Springer-Verlag, New York, 2000.
[21] R. B. Arps, Binary image compression. In Image Transmission Techniques, pages 219 – 276. Academic Press, W. K. Pratt (Ed.), New York, 1979.
[22] H. Samet, Application of spatial data structures: Computer Graphics, Image
Processing and GIS. Addison Wesley, Reading , MA, 1990.
[23] J. A. Storer and H. Helfgott, Lossless image compression by block matching. The Computer Journal, 40:137 – 145, 1997.
[24] G. Langdon and J. Rissanen, Compression of black white images with arithmetic coding. IEEE Transactions on Communications, 29:858 – 867, 1981.
[25] A. Moffat. Two-level context based compression of binary images. In Proc. Of the Data Compression, pages 382 – 391. IEEE Computer Society Press, 1991.
[26] D. Salomon. Data Compression: The Complete Reference. Springer-Verlag, New York, 2000.
[27] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Prentice Hall, 2002.
[28] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations. IEEE Transactions on Image Processing, 1:18 – 30, 1992.
[29] G. E. Oien and S. Lepsoy, Fractal-based image coding with fast decoder convergence. Signal Processing, 40:105 – 117, 1994.

[30] L. Thomas and F. Deravi, Region-based fractal image compression using heuristic search. IEEE Transactions on Image Processing, 4:832 – 838, 1995.
[31] Z. Wang and A. C. Bovik, A universal image quality index. IEEE Signal Processing Letters, 9:81 – 84, 2002.
[32] Z. Wang, A. C. Bovik, and L Lu, Why is image quality assessment so difficult? In Proc. of the IEEE international conference on Acoustic, Speech, and Signal Processing, pages 3313 – 3316, 2002.
[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13:600 – 612, 2004.
[34] E. J. Delp and O. R. Mitchell, Image compression using block truncation coding. IEEE Transactions on Communications, 27:1335 – 1342, 1979.
[35] Jing-Ming Guo & Yun-Fu Liu, Improved Block Truncation Coding Using Optimized Dot Diffusion, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 3, MARCH 2014
[36] Jayamol Mathews, Madhu S. Nair & Liza Jo, Modified BTC Algorithm for Gray Scale Images using max-min Quantizer, IEEE 2013.
[37] Ki-Won Oh & Kang-Sun Choi, Parallel Implementation of Hybrid Vector Quantizer based Block Truncation Coding for Mobile Display Stream Compression, IEEE ISCE 2014.
[38] Jing-Ming Guo, Heri Prasetyo, and Jen-Ho Chen, Content-Based Image Retrieval Using Error Diffusion Block Truncation Coding Features, IEEE Transactions on Circuits and Systems for Video Technology 2014.
[39] Seddeq E. Ghrare & Ahmed R. Khobaiz, Digital Image Compression using Block Truncation Coding and Walsh Hadamard Transform Hybrid Technique, IEEE conference 2014.
[40] Jing-Ming Guo & Heri Prasetyo, Content-Based Image Retrieval Using Features Extracted From Halftoning-Based Block Truncation Coding, IEEE Transactions on image processing, vol. 24, no.3, march 2015.