International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 8 (2015) pp. 18917-18934

http://www.ripublication.com

Block Processed Error Diffusion For High Speed Compression of Still Images

S.Chandravadhana[1] and N.Nithiyanandam [2]
[1]*Department of Electronics and Communication*
Agni College of Technology
Thalambur, Chennai, Tamilnadu
INDIA
chandra.vadhu@gmail.com
[2]*Department of Electronics and Communication*
B.S.Abdur Rahman University
Vandalur, Chennai, Tamilnadu
INDIA
mail_nithi@yahoo.com

Abstract

Various compression techniques for still images are present in the literature. Block Truncation Coding has been considered as the most simplest and efficient form of Image compression technique. However, any attempt to increase the block size for obtaining low bit rates using the traditional Block Truncation Coding, causes blocking artifacts and false contour to dominate the image and also the edges are completely blurred. To overcome the problem of false contour and blurring of the edges an improved form of Block Truncation Coding has been proposed and to ease the blocking artifacts present in the image, a digital halftoning technique namely the error diffusion with a derivative of Floyd Steinberg kernel is proposed. Also a filtering technique of Unsharp Masking is used for sharpening the edges of the image and to improve the peak signal to noise ratio is proposed. This method shows that it is much superior to the traditional Block Truncation Coding and other error diffusion kernels when comparison is made with respect to the Compression Ratio and Peak Signal to Noise Ratio, time of processing and bit rate of the images.

Key-Words: Block Truncation Coding, Bit Rate, Digital Halftoning, Error Diffusion, Floyd Steinberg Kernel, Image Compression, Compression Ratio, Peak Signal to Noise Ratio, bit rate, Unsharp Masking.

Introduction

The method of Block Truncation Coding (BTC) was first introduced by Delp and Mitchell [1], in 1979. This algorithm is based on dividing the image into non overlapping blocks of equal size. Each pixel in a block is then replaced by its high mean or low mean based on the threshold value. The main advantage of this method lies in the lower computational complexity and good image quality [2]-[6] compared to other methods of compression such as JPEG or JPEG 2000. Nevertheless, when the block size is increased for obtaining low bit rates and thereby high compression ratio, the quality of the image degrades rapidly. To improve the quality of the image, several methods have been proposed, such as vector quantization (VQ) which improves the compression ratio [2] ; A hybrid coding method by using look up tables (LUT) and VQ to encode the bit map and low mean of the blocks [3]; Adapting to the universal Hamming Codes and a differential pulse code modulation (DPCM) to the bit plane and the side information of BTC to reduce the bit rate [4]; using moment preservation and visual information to further compress the image and to retain the image quality for real time processing [5]. However all these methods achieve moderate image quality and high compression ratio at the cost of higher computational complexity. In order to reduce the blocking effects of the BTC, some halftoning[16] based BTC methods have been developed to improve the image quality and to reduce the computational complexity. Digital halftoning [7] is a technique for converting gray level images into two tone binary images. These halftone images can resemble the original images when viewed from a distance by the low-pass nature of the Human Visual System (HVS). When HVS is involved, halftoning-based BTC schemes can effectively ease the annoying blocking effect and false contour in the typical BTC. With the inherent low complexity nature of the halftoning-based BTC, it can be easily deployed in low power devices, such as those hand-held devices and low power systems such as image/video recording and satellite image transmission.

Digital halftoning techniques can be further classified according to their performance namely1) dot diffusion [10]-[11], 2) error diffusion [12], 3) ordered dithering [13]-[14] and 4) iterative process - Direct Binary Search (DBS) [15]. Among these, DBS provides the best image quality, yet the processing efficiency is still an issue for its inherent iteration updating strategy. Although in Lieberman-Allebach's work [9], a fast algorithm was proposed to speed up the overall efficiency, the increasing resolution in digital images still makes the DBS impractical. Error diffusion has the second best quality; especially the published Tone-Dependent Error Diffusion (TDED) [9] can achieve similar quality to that of DBS, in which the memory requirement had also been considered [9]. In addition, the issue of processing efficiency was also addressed by utilizing the concept of two non-overlapped blocks to process independently and asynchronously for achieving parallelism. The image quality generated by the ordered dithering is inferior to the above two techniques, yet it can provide the highest processing speed by enjoying its inherent parallelism. In addition, the "dot gain" problem [10]-[11] can be eased by appropriately arranging its Dither Array (DA). Dot diffusion [10] provides a trade-off performance in between image quality and processing efficiency. The rendered blue-noise distribution [7] is similar to that of the error diffusion, while the parallelism is similar to that of the ordered dithering. This paper presents an improved form of BTC along with the derivative of Floyd Steinberg error diffusion called the modified Floyd Kernel in order to improve the compression ratio and Peak Signal to Noise Ratio (PSNR) and to reduce the bit rate and time of processing.

Traditional Block Truncation Coding

Traditionally, Block Truncation Coding [1], divides an image of size, say P x Q into corresponding blocks of size M x N and each block is processed independently. The flow diagram of BTC is shown in Fig 1. The traditional BTC computes the first moment (mean), second moment and the variance as shown in (1), (2) and (3)

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_{i,j} \tag{1}$$

$$\bar{x}^2 = \frac{1}{m}\sum_{i=1}^{m} x_{i,j}^{\;2} \tag{2}$$

$$\bar{\sigma}^2 = \left| \bar{x}^2 - \left| \bar{x} \right|^2 \right| \tag{3}$$

Where is the gray scale value of the pixels in the original image. The concept of BTC is $x_{i,j}$ to preserve the first and the second moment characteristics of a block when the original block is substituted by its quantization levels. Thus the high and low mean of the individual blocks are calculated as shown in equations (4) and (5).

$$a = \bar{x} + \bar{\sigma}\sqrt{\frac{m-q}{q}} \tag{4}$$

$$b = \bar{x} - \bar{\sigma}\sqrt{\frac{q}{m-q}} \tag{5}$$

Where *a* and *b* denote the high and low mean of the blocks, *q* denotes the number of pixels greater than the mean , and *m* is the total number of pixels in the block respectively. The bit plane is then obtained by taking the mean as the threshold. Therefore $\bar{x}$ BTC is also called as one bit quantizer. Here the bit plane is used to record the distributions of the two quantization levels, the high mean and the low mean.

$$y_{i,j} = \begin{cases} a, if \ h_{i,j} = 1, \\ b, if \ h_{i,j} = 0, \end{cases} where \ h_{i,j} = \begin{cases} 1, if \ x_{i,j} \geq \bar{x}, \\ 0, if \ x_{i,j} < \bar{x} \end{cases} \tag{6}$$

Where is the individual element in the compressed block and $h_{i,j}$ $y_{i,j}$ is the element of the bit plane. The binarized bit plane is transmitted along with the two quantization levels through the channel. Therefore the bit rate of a BTC compressed image is calculated as shown in equation (7).

Bit Rate = (M x N x 1) + (8 x 2) (7)

Where M x N denotes the number of elements in an individual block. Since the traditional BTC exhibits annoying blocking artifacts and false contour for higher block sizes, an improved form of BTC is proposed. This is then combined with a modified form of Floyd kernel for error diffusion to ease the blocking artifacts and to improve the Peak Signal to Noise Ratio (PSNR) of the image.

Improved Block Truncation Coding

In the traditional BTC, as we increase the block size for obtaining high compression ratio and low bit rate, the image quality is degraded and the false contours dominate the image. This is because of the fact that the first and second moments should be preserved in the compressed image. The calculation for low mean and high mean for the blocks in the image are carried out as shown below.

Therefore in our work, an improved formula is incorporated for the values of high mean and low mean in each block. This value of high mean and low mean produce sharp edges and also satisfactory compression ratio in the retrieved image with low channel bit rates. The following equations (8) and (9) show the improved formula for the high mean and low mean respectively.

$$a = (\bar{x} + Maxvalue) / 2 \tag{8}$$

$$b = (\bar{x} + Minvalue) / 2 \tag{9}$$

Where '*Maxvalue*' denotes the maximum value of the pixel in the block and '*Minvalue*' denotes the minimum value of the pixel in the block. This formula does not require the calculation of variance and second moment. Therefore low bit rates and satisfactory compression ratio can be achieved by applying this formula. The computational complexity has also been greatly reduced here. The comparison of computational complexities between the traditional and the improved BTC is shown in Table 1. Figure 2 shows the complete compression algorithms using the Traditional BTC and Improved BTC algorithm. This algorithm works well with weather satellite images (Visible, Infra-Red and Water Vapour), aerial images and standard images. Three sample images have been taken for analysis and the significant improvement in the parameters such as PSNR and MSE of the images are measured.

Table 1: Comparison of computational complexities of the traditional BTC and the improved BTC

Algorithm	*No. of additions/ Subtractions*	*No. of Multiplications/ division*	*Square root operations*
Traditional BTC	[2(M x N)] + 3	[M x N] +9	2
Improved BTC	[M x N] + 2	[M x N] + 2	0

The bit rates for various block sizes of the Block Truncated Image are shown in Table 2.

Table 2: Bit Rates for various Block Sizes of BTC

Block Size	*Bit Rate for Traditional BTC / Improved BTC (bits per pixel)*
4 x 4	2
8 x 8	1.125
16 x 16	1.0625
32 x 32	1.0156

In measuring the quality of an image, the Peak Signal to Noise Ratio (PSNR) plays an important role. Also, our main aim here is to reduce the bit rate and achieve satisfactory Compression Ratio (CR). Equations (10) and (11) give the measure of the CR and PSNR of an image.

$$CR = \frac{\text{No. of bits used to represent the original image}}{\text{No. of bits used to represent the compressed image}} \tag{10}$$

$$PSNR = 10\log_{10}\left(\frac{P \times Q \times 255^2}{\sum_{i=1}^{P}\sum_{j=1}^{Q}\left[\sum_{m,n\in R}\sum w_{m,n}\left(g_{i+m,j+n} - h_{i+m,j+n}\right)\right]^2}\right) \tag{11}$$

Traditional BTC and Error Diffusion

Since BTC exhibits annoying false contour, a digital half toning technique known as Error diffusion [12] is combined with BTC in order to ease the false contour. This is done by diffusing the quantization errors to the neighbouring pixels. In this process the pixels in each block of the image is raster scanned from left to right and top to bottom. Here the process is done based on the high correlation between the adjacent pixels. The error diffusion block diagram is shown in Figure 1.

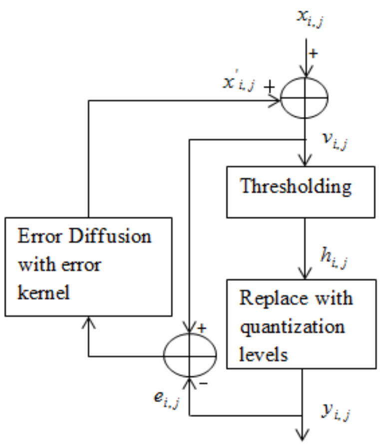

Figure 1: Flow diagram of Error Diffusion and BTC

The following equations depict the error diffusion process for the quantized block of BTC.

$$v_{i,j} = x_{i,j} + x'_{i,j}, \tag{12}$$

$$\text{where} \quad x'_{i,j} = \sum_{m,n} e_{i+m,\,j+n}\, ek_{m,n}$$

$$e_{i,j} = v_{i,j} - y_{i,j} \tag{13}$$

$$\text{where} \quad y_{i,j} = \begin{cases} a, & \text{if } h_{i,j}=1 \\ b, & \text{if } h_{i,j}=0 \end{cases} \quad . \text{ and}$$

$$h_{i,j} = \begin{cases} 1, & \text{if } v_{i,j} \geq \bar{x} \\ 0, & \text{if } v_{i,j} < \bar{x} \end{cases}$$

Here, the gray scale values of the original image are denoted by and the error diffused output is substituted as which is replaced by the high mean and the low mean after error diffusion. In the work related to BTC and error diffusion [17]-[18], the kernels used for error diffusion are shown in Figure 2.

(a)

	*	7/16
3/16	5/16	1/16

(b)

		*	7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

(c)

		*	8/42	5/42
2/42	4/42	8/42	4/42	2/42
1/42	2/42	4/42	2/42	1/42

Figure 2: Three well known Error diffusion kernels: (a) Floyd and Steinberg (b) Jarvis (c) Stucki

Where '*', denotes the current pixel to be processed. The Jarvis and Stucki kernels used for error diffusion needs 12 neighbouring pixels to diffuse the errors and thus require more time to execute the algorithm. Since the essential issue in this study is the time factor, we do not consider the two kernels. The results after Error Diffused Block Truncation coding using the Floyd and Steinberg kernel[8], shows that the blocking artifacts and false contour are eased when the error diffusion kernel is applied to the Block Truncated image [19]. The comparison of the original image with BTC and EDBTC Floyd and Steinberg kernel is shown in Figure 2.

The error diffused BTC images show that though the false contour is eased, there exists a wormy effect. As the block size increases, the visual quality of the image reduces. The measure of the PSNR for image lena. jpg are compared and tabulated in Table 3.

Table 3: Comparison of PSNR values of BTC and Error diffused BTC

Traditional BTC (Block Size)	*Peak Signal to Noise Ratio (PSNR)*	Error Diffused BTC (Block Size)	*Peak Signal to Noise Ratio (PSNR)*
4 x 4	47.93	4 x 4	49.55
8 x 8	48.65	8 x 8	49.62
16 x 16	48.98	16 x 16	50.23
32 x 32	49.45	32 x 32	50.84

In order to increase the speed of execution of the block processed error diffusion algorithm and to improve the PSNR values with satisfactory compression ratio, a modified form of the Floyd and Steinberg kernel is used for diffusing the quantization errors of the improved BTC. This method is described in the next section.

Improved BTC and modified Error Diffusion Kernel

The error diffusion kernel used for the traditional BTC namely the Floyd and Steinberg has the disadvantage of low PSNR and worming effect. In order to avoid this, the improved BTC is used along with a modified Floyd and Steinberg kernel. Diffusing the errors into this kernel gives a satisfactory PSNR and CR values for the image [19]. Figure 3 shows the modified Floyd kernel.

$$\begin{array}{cccc} & * & 1/8 & 1/8 \\ 1/8 & 1/8 & 1/8 & \\ & 1/8 & & \end{array}$$

Figure 3: Modified Error diffusion Kernel

The values in the error diffusion kernel is chosen based on the correlation between the neighbourhood pixels and the direction in which the errors are diffused. This kernel is applied to the improved BTC image for various block sizes. The equations for the pixels to be processed in the modified error diffusion kernel are shown from (14) to (19).

$$x_{i.j+1} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i,j+1} \tag{14}$$

$$x_{i.+1,j} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i+1,j} \tag{15}$$

$$x_{i.+1,j-1} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i+1,j-1} \tag{16}$$

$$x_{i.+1,j+1} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i+1,j+1} \tag{17}$$

$$x_{i.j+2} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i,j+2} \tag{18}$$

$$x_{i.+2j} = \left[\frac{1}{8} * e_{i,j}\right] + x_{i+2,j} \tag{19}$$

The error diffused image is then filtered using an unsharp masking filter[20] for edge enhancement.. The procedure to perform Unsharp masking is shown in section 5.1.

Unsharp Masking

The following are the steps to be followed for Unsharp Masking:

a) Blur the image using a Gaussian filter and compare it with the original image.
b) If the difference is greater than a user-specified threshold setting, the images are (in effect) subtracted.
c) Multiply the result obtained in (b) by some weighting fraction.
d) Add the result obtained in (c) to the original image.

Mathematically, the Unsharp masking filter is represented as shown in (20).

$$f'(m,n) = f(m,n) + \alpha[f(m,n) - \bar{f}(m,n)] \tag{20}$$

Where $\bar{f}(m,n)$ is the original image, $f(m,n)$ is the blurred version of the original image, is α the weighting fraction and the term $f'(m,n)$ is the sharpened result. The results after Unsharp mask filtering depicts that the sample image 3 gives superior sharpening values.

Experimental Results

In the present work, the PSNR, the Mean Square Error(MSE) and the processing time are calculated for three test images and compared with the traditional BTC. Further, the Floyd and Steinberg error diffusion kernel is compared with the modified Floyd and Steinberg error diffusion kernel in terms of PSNR and MSE. This algorithm is applied to higher block sizes of the images, which improves the compression ratio and thereby reduces the transmission bit rate. It is found that the compression ratio is satisfactory as the block size increases and also, the PSNR values are obtained for higher block sizes of the images is improved. The time taken to execute the proposed algorithm is also reduced thereby enabling the transmission of more number of images from the transmitter to the receiver. The bit rates are also reduced for higher block sizes. The higher block size images do not show any blocking artifacts as is the case of the Traditional BTC. Also the blurred edges which are inherent in the Traditional BTC are also removed by this proposed algorithm. The following Figure 5 shows the comparison of PSNR values for traditional BTC and improved BTC for the image lena.jpg

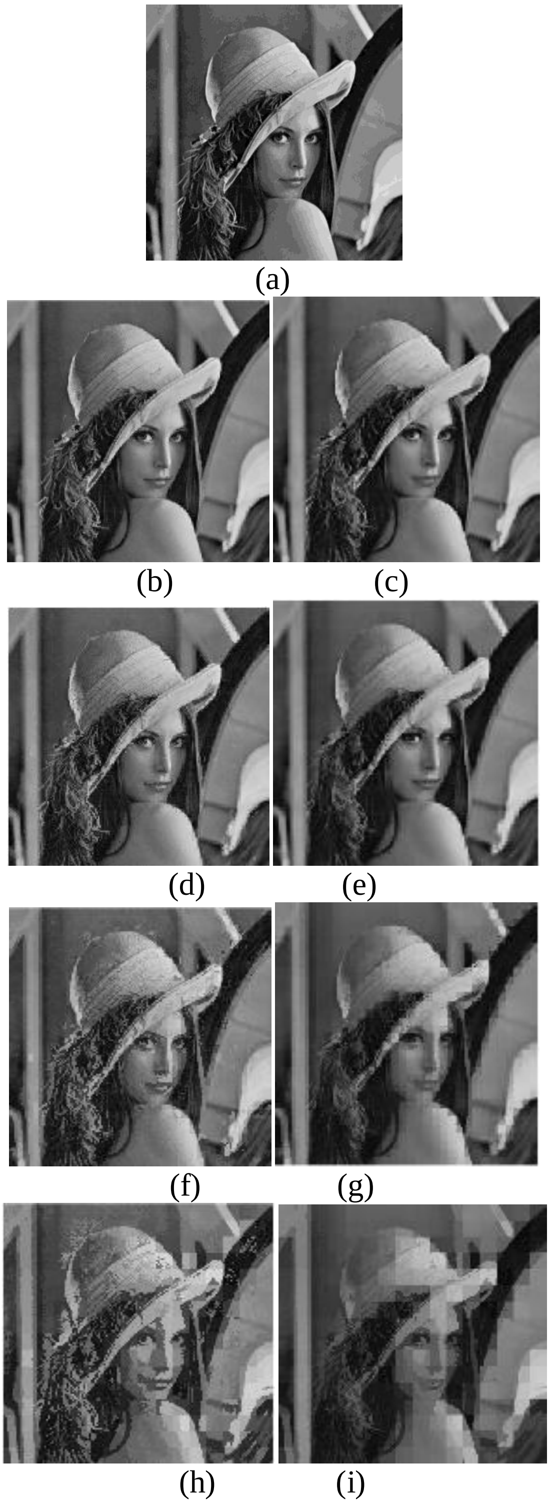

Figure 4: (a) Original image lena. jpg (b),(d),(f)and (h)- Traditional BTC for block size 4x4,8x8,16x16 and 32x32. (c),(e),(g) and (i) – improved BTC for block size 4x4,8x8,16x16 and 32x32.

Table 4: Comparison of PSNR for traditional BTC and improved BTC

Block Size	*PSNR for traditional BTC*	*MSE for traditional BTC*	*PSNR for improved BTC*	*MSE for improved BTC*
4 x 4	47.93	0.59	50.12	0.48
8 x 8	48.65	0.56	50.55	0.46
16 x 16	48.98	0.51	51.60	0.45
32 x 32	49.45	0.50	51.87	0.43

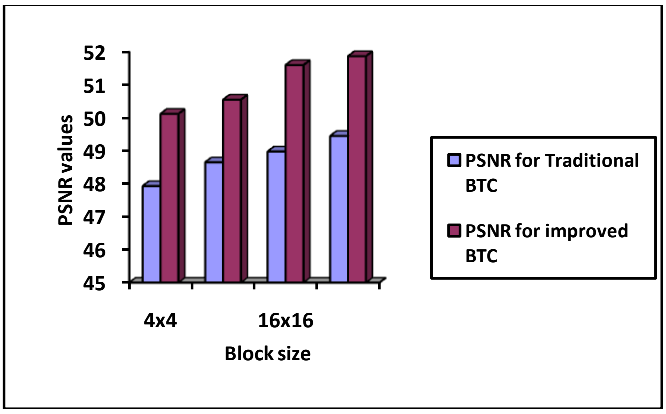

Figure 5: Graph showing the comparison of PSNR values for the traditional BTC and the improved BTC.

In the Figure 6, the Mean Square Error values are plotted for various block sizes and comparison is made for the traditional and improved BTC.

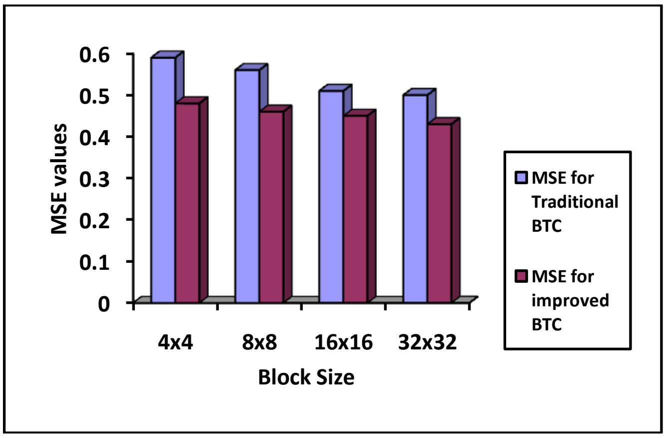

Figure 6: Graph showing the comparison of MSE values for the traditional BTC and the improved BTC.

The improved method of BTC takes lesser time for computation as its computational complexity is lower compared to the traditional BTC. The following

Table 5 shows the comparison of processing time for the two methods of BTC for the image lena.jpg.

Table 5: Comparison of CPU processing time for traditional BTC and improved BTC

Block Size	***CPU processing time for traditional BTC(in seconds)***	***CPU processing time for improved BTC(in seconds)***
4 x 4	3.24	1.35
8 x 8	1.48	0.94
16 x 16	0.87	0.68
32 x 32	0.59	0.49

The graph showing the comparisons of the execution times of the two BTC methods in shown in Figure 7.

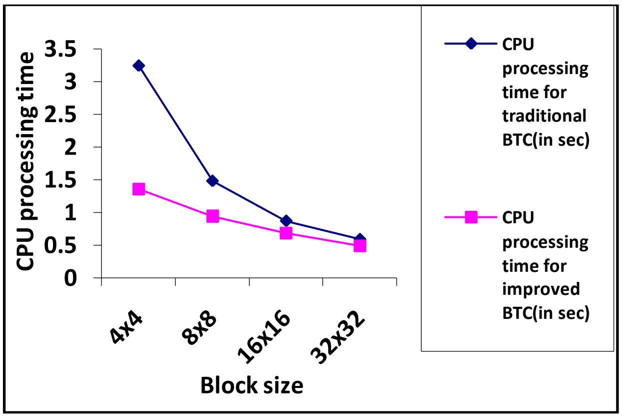

Figure 7: Comparison of CPU processing time for traditional BTC and improved BTC

After the application of BTC, the Floyd and Steinberg error diffusion kernel is applied to the image boat.jpg and the results are tabulated. Also, the modified kernel of Floyd and Steinberg is applied to the improved BTC image (boat.jpg) and the results are compared for the PSNR values. The following Figure 8 shows the comparison of both the methods. The Table 6 shows the comparison of PSNR values for the two methods of error diffusion.

(a)

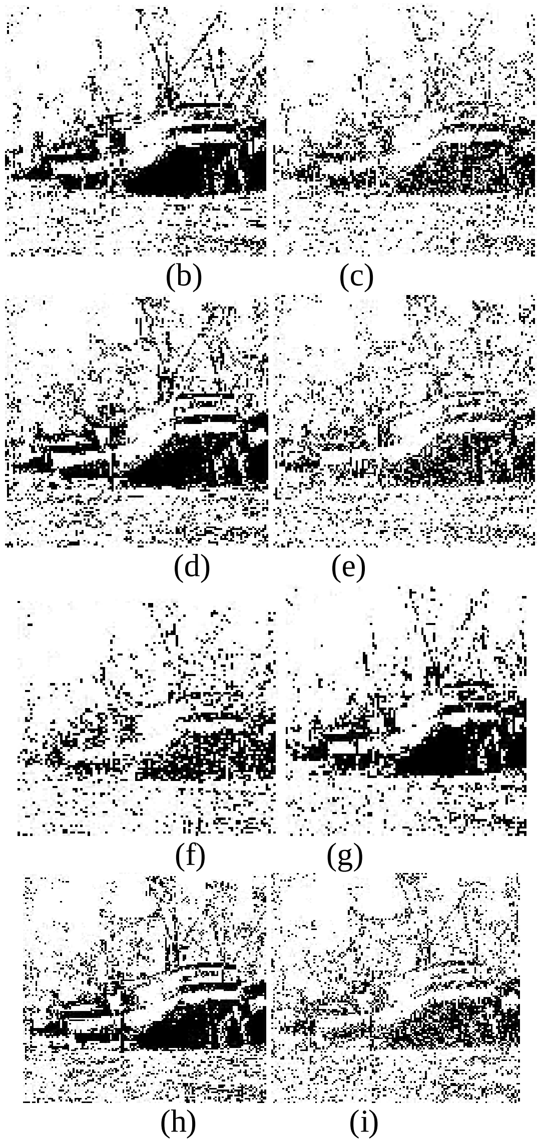

Figure 8: (a) Original image boat. jpg, (b),(d),(f) and (h)-Traditional BTC and error diffusion for block size 4x4,8x8,16x16 and 32x32. (c),(e),(g) and (i) improved BTC and modified error diffusion for block size 4x4,8x8,16x16 and 32x32.

Table 6: Comparison of Error diffused BTC and modified error diffused BTC.

Block Size	***PSNR for Traditional BTC and Floyd-Steinberg error kernel***	***PSNR for improved BTC and modified Floyd-Steinberg error kernel***
4 x 4	48.67	50.75
8 x 8	48.97	51.24
16 x 16	49.02	51.68
32 x 32	49.52	51.98

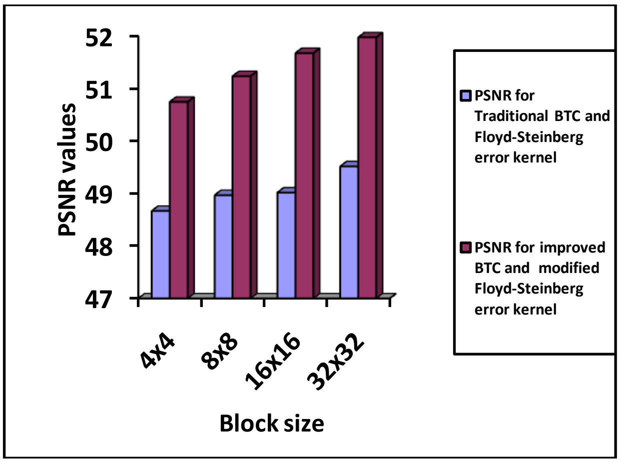

Figure 9: Graph showing the comparison of PSNR for error diffused BTC and modified error diffused BTC.

The improved BTC and the modified Floyd and Steinberg error diffusion kernel, when applied to boat.jpg, reduce the blocking artifacts, and also an improvement in the PSNR values are obtained for various block sizes. In order to enhance the image further, Unsharp masking is applied. The following Figure 10 shows Unsharp masking applied to the improved BTC and modified error diffusion to the image barbara.jpg for various block sizes. The images show that an enhanced version of the original image barbara.jpg is obtained after the improved BTC and modified error diffusion kernel is applied and followed with unsharp masking technique.

(a)

(b) (c)

(d) (e)

Figure 10: (a) Original image, (b), (c),(d) and (e) –Unsharp masking applied to barbara.jpg after applying improved BTC and modified error diffusion kernel for block size 4x4,8x8,16x16 and 32x32.

Table 11: Comparison of PSNR values between improved BTC-modified error diffusion kernel and Unsharp masking.

Block Size	*PSNR for improved BTC and modified Floyd-Steinberg error kernel*	*PSNR for improved BTC and modified Floyd-Steinberg error kernel and Unsharp Masking*
4 x 4	50.75	51.05
8 x 8	51.24	51.56
16 x 16	51.68	51.81
32 x 32	51.98	52.49

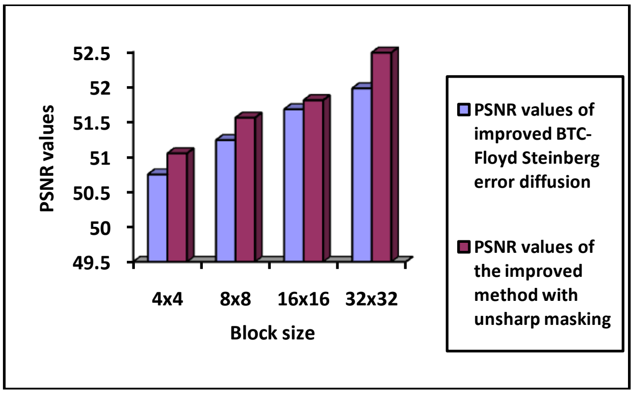

Figure 11: Graph showing the comparison of the improved method of BTC-error diffusion and Unsharp masking.

Sharpness is one of the quality measures for an image. This parameter gives the measure of finest details in the image. Therefore we apply the unsharp mask filter to the modified error diffused image Barbara.jpg. The sharpness of the image is measured after it is filtered using Unsharp Masking filtering. A 3x3 mask is used and the results for the sharpness of the original (improved BTC and modified error diffusion) image. The filtered image with the unsharp mask filter is better than other filters such as Mean filter. The results show that the unsharp masking filter produces sharp edges. The sharpness of the image is measured in Cycles / Pixel for a kernel of size 3x3. Table 12 depicts the sharpness comparisons of the modified error diffused image and the filtered image using unsharp mask.

Table 12: Comparison between sharpness values of the modified Floyd error diffused image and filtered image using unsharp mask for barbara.jpg.

BTC Block Size	*Sharpness of the improved BTC and modified Error Diffusion*	*Sharpness of the image after applying Unsharp masking filter*
4 x 4	14.415	47.142
8 x 8	15.952	52.077
16 x 16	17.866	58.204
32 x 32	19.183	62.273

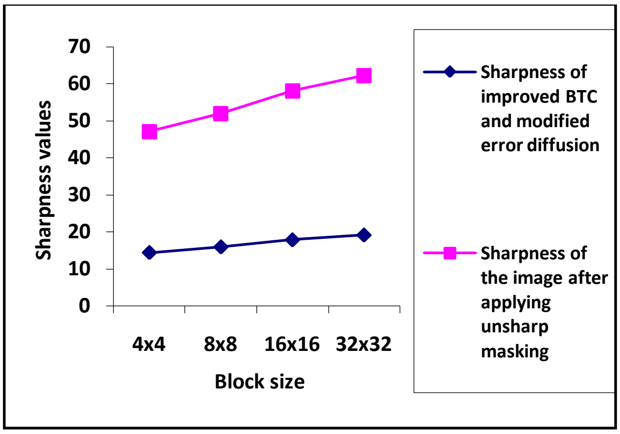

Figure 12: Graph showing the sharpness values of the modified Floyd error diffused image and filtered image using unsharp mask.

Table 13: Execution Time of the CPU

Block Size	*Traditional BTC and Error Diffusion(time in sec)*	*Improved BTC and modified Error diffusion(time in sec)*
4 x 4	4.20	3.94
8 x 8	1.56	1.32
16 x 16	0.89	0.79
32 x 32	0.76	0.70

The Figure 13 below shows the comparison of the execution time for traditional BTC and error diffusion and improved BTC and modified error diffusion.

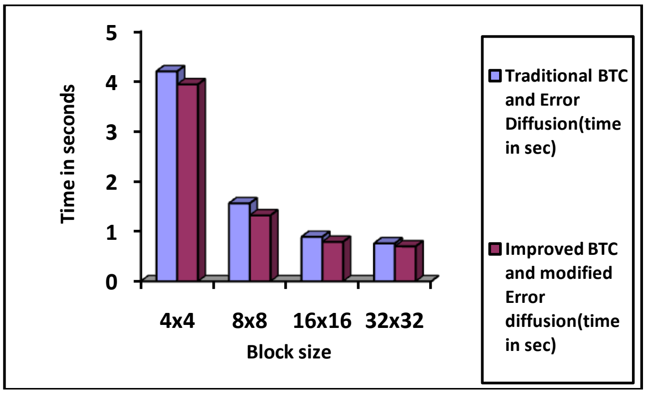

Figure 13: Block Size Vs Execution time

Conclusions

In this paper, a simple and efficient algorithm namely the Block Truncation Coding is taken for compressing still images. The traditional Block Truncation Coding has got the disadvantage of producing severe blocking artifacts when the block size is increased for obtaining higher compression ratio and lower bit rates. In order to decrease the time of processing and to reduce the computational complexity, an improved method is developed. This method produces sharp edges and thereby the blurred edges are reduced. Further to this an error diffusion method is carried out with a modified form of Floyd and Steinberg error diffusion kernel. This improved kernel removes the blocking artifacts and produces better PSNR than the original Floyd and Steinberg kernel. The methodologies are applied to three standard test images and satisfactory results are obtained. This method can also be applied to satellite images where large numbers of images are transmitted to the earth station and efficient usage of bandwidth is required.

References

[1] E. J. Delp and O. R. Mitchell, “Image compression using block truncation coding,” IEEE Transactions on Communication Systems., vol. 27, no. 9, pp. 1335–1342, Sep. 1979.

[2] V. R. Udpikar and J. P. Raina, “BTC image coding using vector quantization,” IEEE Transactions on Communications, vol. 35, no. 3, pp. 352–356, Mar.1987.

[3] Y. Wu and D. C. Coll, “BTC-VQ-DCT hybrid coding of digital images,” IEEE Trans. Commun., vol. 39, no. 9, pp. 1283–1287, Sep.1991.

[4] C. S. Huang and Y. Lin, “Hybrid block truncation coding,” IEEE Signal Process. Lett., vol. 4, no. 12, pp. 328–330, Dec. 1997.

[5] Y.-G. Wu and S.-C. Tai, “An efficient BTC image compression technique,” IEEE Trans. Consumer Electron., vol. 44, no. 2, pp. 317–325, May 1998.

[6] Y. C. Hu, “Improved moment preserving block truncation coding for image compression,” Electron. Lett., vol. 39, no. 19, pp. 1377–1379, Sep. 2003.

[7] R. Ulichney, Digital Halftoning. Cambridge, MA: MIT Press, 1987.

[8] R. W. Floyd and L. Steinberg, “An adaptive algorithm for spatial gray scale,” in Proc. SID Dig. Soc. Inform. Display, 1975, pp. 36–37.

[9] J. F. Jarvis, C. N. Judice, and W. H. Ninke, “A survey of techniques for the display of continuous-tone pictures on bilevel displays,” Comput Graph. Image Process., vol. 5, no. 1, pp. 13–40, 1976.

[10] D. E. Knuth, “Digital halftones by dot diffusion,” ACM Trans. Graph.,vol. 6, no. 4, pp. 245–273, Oct. 1987.

[11] M. Mese and P. P. Vaidyanathan, “Optimized halftoning using dot diffusion and methods for inverse halftoning,” IEEE Trans. Image Process., vol. 9, no. 4, pp. 691–709, Apr. 2000.

[12] J. M. Guo, “Improved block truncation coding using modified error diffusion,” IET Electron. Lett., vol. 44, no. 7, pp. 462–464, Mar.2008.

[13] J. M. Guo and M. F. Wu, “Improved block truncation coding based on the void-and-cluster dithering approach,” IEEE Trans. Image Process., vol. 18, no. 1, pp. 211–213, Jan. 2009.

[14] J. M. Guo, M. F. Wu, and Y. C. Kang, “Watermarking in conjugate ordered dither block truncation coding images,” Signal Process., vol. 89, no. 10, pp. 1864–1882, 2009.

[15] Jing-Ming Guo, Chang-Cheng Su: Improved Block Truncation Coding Using Extreme Mean Value Scaling and Block-Based High Speed Direct Binary Search. IEEE Signal Process. Lett. 18(11): 694-697 (2011).

[16] Yun-Fu Liu, Jing-Ming Guo, and Jiann-Der Lee, “Halftone Image Classification Using LMS Algorithm and Naive Bayes”, IEEE Trans. Image Process., vol. 20, no. 10, pp. 2837–2847, Oct. 2011.

[17] Sarailidis.G, Katsavoudinis.I,”A Multiscale Error Diffusion Technique for Digital Multitoning”, IEEE Trans. Image Process., vol. 21, no. 5, pp. 2693–2705, Jan 2012.

[18] Jing-Ming Guo and Yun-Fu Liu,” High Capacity Data Hiding for Error-Diffused Block Truncation Coding”, IEEE Trans. Image Process., vol. 21, no. 12, pp. 4808–4818, Dec 2012.

[19] The Image database available online at www.imd.gov.in/ section/satmet/dynamic/ insat sector.

[20] Li Wang, Defeng Wang, Lin shi and Chu.W.C," Radiographs enhancement based on unsharp masking and Gray-level Grouping," IEEE International Conference on Biomedical Health and Informatics, pp -350-353, Jan 2012.