Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.2, 2019, pp.90-97
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2019.8.2.2134

An Enhanced Technique for Fractal Image Compression Using Block Based Features (BBF)

Dr. T. Velumani[1], A. R. Karthekeyan[2] and R. Saveetha[3]
[1,2&3]Assistant Professor, Department of Computer Science,
Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India
E-Mail: velumani46@gmail.com, karthikmca.srkv@gmail.com, mailsavee@gmail.com

(Received 3 April 2019; Revised 17 April 2019; Accepted 30 April 2019; Available online 5 May 2019)

***Abstract* - Image Compression is very active research area specializing on how to compression and decompression of images. The various methods have been proposed for fractal image compression techniques for storage and drawbacks. The complexity in process will affect performance of the existing system to make insufficient. In this paper, the proposed research work presented a block based feature compression of image in the generation of feature sets. The feature sets are generated based such as vertical, horizontal and are extracted from the image based on range and domain blocks. The extracted features are carried out with encoding and decoding the images. The experimental result shows that block based feature method provides better compression ratio and other parameters when compared with the existing methods such as BTC, Wavelet and DCT.**

***Keywords:* Fractal Image Compression, Range and Domain Blocks, Encoding and Decoding the Images, BTC - Wavelet and DCT**

I. INTRODUCTION

Compression is the process of reducing the size of a file via encoding its information records extra correctly. In this approach to lessen the variety of bits and bytes are used to keep the information. There are two types of image compression are Lossy Compression and Lossless Compression. To obtain the excellent possible reliability for the given bit rate to attain a given fidelity measure. It reduces a data by permanently putting off certain data especially redundant facts. The data is compressed and can be reconstituted without loss of information noted reversible compression structures. Lossless compression is used whilst each bit of statistics is wanted in the long run product, frequently whilst transmitting a report to a designer. The advance in this data era required a mass facts storage and rapid communication. Storing of images in much less memory leads to a right away reduction within the storage value and quicker facts transmissions. To validate the efforts in non-public sectors require a new picture compression algorithms. Images are stored on computer systems as collections of bits representing pixels.

II. LITERATURE SURVEY

MiarNaimi *et al.*, (2007) [22] published in which the time of encoding process is considerably reduced. In this approach only domain blocks with entropies greater than a threshold are considered.

Mehdi Salarian *et al.*, (2008) [4] presented fractal image compression that has some desirable properties like high quality at high compression ratio, fast decoding and resolution independence. It suffers from long encoding time due to its need to find the best match between sub blocks. This time is related work is used and fast encoding algorithm based on without any search method. To increase a more blocks are covered in initial step of quad tree algorithm.

Chakrapaniand *et al.*, (2009) [5] technique of Genetic Algorithm is applied for Fractal Image Compression with help of this evolutionary algorithm effort is made to reduce the search complexity of matching between range block and domain block. The image compression techniques in spatial domain are fractal image compression but the main drawback involves more computational time due to global search. In order to improve the computational time and quality of the decoded image.

Anupam Garg (2011) [8] proposed method for image compression has which uses fractal image compression encoding algorithm with thresholding and then fractal decoding algorithm to get decompressed image. The fractal based image compression has the disadvantage of having long encoding time.

Sachin Dhawan (2011) [6] discussed about the image compression, need of compression with various algorithm. It attempts to give a recipe for selecting one of the popular image compression algorithms based on Wavelet, JPEG/DCT, VQ and Fractal approaches.

Sumathi Poobal *et al.*, (2011) [7] Quad-Tree partitioning method of Fractal Image Compression, a lossy image compression method which works on self-similarity is applied on different imaging modalities like X-ray, CT scan, angiogram, mammogram and ultrasound. The quality of the compressed images becomes an important factor along with considerable compression ratio determines effective image compression.

Chetan Dudhagara (2012) [9] proposed on fractal-based image compression and fixed-size partitioning will be analyzed for performance and compared with a standard frequency domain based image compression standard JPEG.

It performs compression and decompression used for simple images. In Fractal Image Compression, when we increase the panel size, it will decrease the original file size and increases the image compression ratio and also decrease the processing time.

Venkatasekhar (2012) [10] proposed genetic algorithm which is used to find the best block of replacement, so fractal image is done easily. Intuitively, Huffman coding can offer better fast fractal compression than Arithmetic coding. Whenever compared to Arithmetic coding, Huffman coding is best for compression. It increases the speed of compression and produces high PSNR. The genetic algorithm increases the speed of convergence for reaching the best block.

Chandan Singh Rawat *et al.*, (2013) [11] presented a large amount of data is necessary to represent the digital images so the transmission and storage of such images are time-consuming and infeasible. In order to overcome this difficulty and to compress the image efficiently, a combination of DCT and fractal image compression techniques is proposed.

Jianji Wang *et al.*, (2013) [12] discussed fractal image compression which is an image coding technology based on the local similarity of image structure. Fractal image compression suffers from the high computational complexity in encoding. In this work, a new fractal image compression scheme is proposed based on the fact that the affine similarity between two blocks in fractal image compression is equivalent to the absolute value of Pearson's correlation coefficient between them to speed up the encoding process while preserving the reconstructed image quality well.

Jie He *et al.*, (2014) [13] presented a sampling set of rules to avoid the need for recoding while magnifying a reconstructed image after fractal image compression. The method put forward in this research work can amplify the software scope of fractal image compression by means of successfully warding off the extra time intake attributable to re-interpreting.

Krishnamoorthy (2014) [14] developed a traditional fractal image compression; the encoding procedure is time-ingesting because of the whole search mechanism. In order to speed up the encoder they undertake particle swarm optimization technique done below type and Dihedral transformation to in addition decrease the amount of MSE computations best 4 adjustments for every domain block are considered to reduce the encoding time.

Preeti Banerjee *et al.*, (2014) [16] novel approach for green fuzzy logic based totally fractal image compression in DCT domain using quad tree algorithm is proposed. It offers with the fuzzy logic primarily based method due to the fact, fuzzy logic is a strong device to deal with vagueness and due to the fact that image are indistinct in phrases of pixel values fuzzy good judgment is maximum appropriate good judgment for its analysis. Within the proposed approach one area block is considered for every variety block and searched simplest for matched assessment scaling. Therefore the results fractal code does now not incorporate coordinates of the matched area block.

Raja Kumari *et al.*, (2014) [15] proposed fractal image compression using discrete wavelet remodel is observed by way of Huffman Run period Encoding. The principle idea is method for both Encoding procedure and picture compression is performed.

Veenadevi *et al.*, (2014) [17] developed a image is partitioned with the aid of thinking about most and minimum length of the range block and transforming the RGB coloration image into YUV shape. Affine transformation and entropy coding are carried out to gain fractal compression.

Nadira Banu Kamal (2015) [18] discussed on iteration free fractal image compression for shade images using the strategies Vector Quantization, Genetic algorithm and Simulated Annealing is proposed for lossy compression, to enhance the decoded image quality, compression ratio and discount in coding time. Usage of synthetic codebook for encoding the usage of Fractal does not require new release at decoding and the coding mistakes are decided right away on the encoder. Only the encoding consumes more time however the deciphering may be very fast.

Priyadarshini *et al.*, (2015) [20] presented survey on the parallel computing of JPEG and Fractal picture compression algorithms. But, most of the image compression strategies have troubles in computational complexity, load and so forth. Parallel computing can efficaciously enhance the processing speed. To speed up the compression process the sequential fractal image compression algorithm wishes to be transformed into parallel fractal image compression algorithm; this translation exploits the inherently parallel nature.

Salarian *et al.*, (2015) [19] proposed in which the time of encoding procedure is appreciably decreased. The algorithm exploits a site pool reduction technique, along with using modern predefined values for contrast scaling factor as opposed to looking it. Simplest the domain blocks with entropy greater than a threshold are considered as domain pool.

Thai Nam Son *et al.*, (2015) [21] advanced an efficient FIC approach applied to colour images, which makes use of a fractal coding on RGB to YUV colour transformation, suitable sampling modes, and special strategies to decrease encoding time. Parallel techniques with the intention to accelerate this section, whilst nonetheless preserve the features of the approach.

III. THEORETICAL BACKGROUND

A. Block Truncation Coding (BTC) Method: Block truncation coding is a easy and speedy lossy compression technique for grey scale images evolved via Delp and Mitchell [1979] [2] [3]. The original of BTC preserves the usual mean and standard deviation. Block truncation coding works by way of dividing the image into in addition to small sub blocks of size 2×2 pixels and then lowering the range of grey levels within each block [2009]. The basic concept of BTC divides the entire image into ‘n’ blocks and codes every block the use of a two-level quantizer. The 2 level degree a and b are selected using the mean $\bar{X}$ and standard deviation (σ) of the grey levels inside block and are preserved. The $\bar{X}$ and σ are calculated using (1) and (2)

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

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

In inverse BTC, two statistical moments a and b are computed the usage of (3) and (4) and are preserved together with the bit plane for reconstructing the image.

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

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

Where q is the variety of pixel values extra than or same to $\bar{X}$ and (m-q) is the quantity of pixels whose gray levels are much less than $\bar{X}$ while reconstructing the image, the ‘o’ within the bit plane is replaced via a and the ‘1’ inside the bit is changed by b.

B. Discrete Cosine Transform DCT Based Transform Coding: Discrete Cosine Transform become first implemented to image compression in the work by means of Ahmed et al [1974]. [1] DCT is a transform coding approach comprising four steps. The foundation image is first partitioned into sub-blocks of size 8x8 pixels in size. Then every block is converted from spatial domain to frequency area the use of a 2-D DCT foundation feature. The ensuing frequency coefficients are quantized and in the end output to a lossless entropy coder. DCT image compression method when you consider that it may decorrelate pixels in the image and compact most image power to few transformed coefficients. BMP and TIFF may be changed by means of Wavelet-primarily based image compression algorithms have better compression overall performance.

C. Proposed Block Based Fractal Image Compression: Fractal Compression was first promoted by M. Barnsley, who founded a company based on fractal image compression technology but who has not released details of his scheme. The first public scheme due to E. Jacobs and R. Boss of the Naval Ocean Systems Center in San Diego who used regular partitioning and classification of curve segments in order to compress random fractal curves in two dimensions.

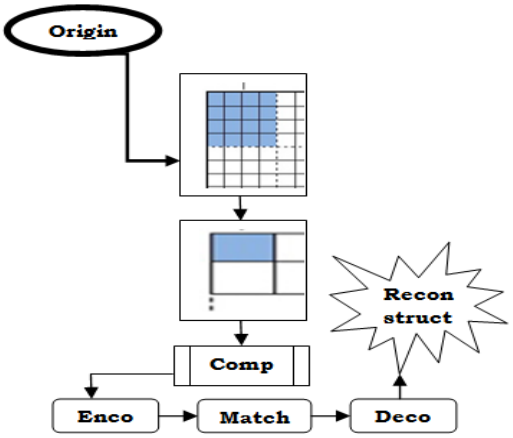


Fig. 1 Process Flow of Encoding & Decoding Block

1. Partitioning Methods: The partitioning scheme used to demarcate the variety of range blocks is one of the most critical factors of the fractal compression method. The constancy and fine of their constructed image, the length and the structure of the fractal code, the shape of the alterations used to map domains into ranges and their descriptions within the fractal, code compression ratio, encoding time and all other crucial characteristics of the compression approach are come what may prompted by using the choice of the partitioning approach.

2. Uniform Partitioning: The uniform partitioning method is not the only option in fractal compression. The uniform partitioning is image-independent because the ranges and domains have fixed size of range is 8×8 (this means that the size of each domain is 16×16). These partitioning methods have drawbacks. First it deals with smaller than the size of the range may be found in the image and then sort of details will be lost during encoding because it would be hard to find a domain with exactly the same details. It found a domain for each range but another problem occurs here there is no certainty that the distance between these two squares will be really small. The size of the ranges can be adjusted to minimize the problem of matching ranges and domains. However, the smaller the ranges are the worse the compression ratio is because the transformations have to be found for larger number of ranges. At the same time, some parts of the image could be covered with larger ranges and the loss of information will be still on acceptable level. This would result in lower number of transformation and better compression ratio would be achieved.

3. Overlapped Range Blocks: The modification of partitioning into squares by this method is very similar to uniform partitioning all ranges have same size *b* × *b* and domains 2*b* × 2*b*. The difference in ranges is not disjunctive but mutually overlapping with half of their size. It means all pixels belong to more than one range pixels close to the edge of the image belong to two ranges and the rest of the pixels are within four ranges. Partitions are encoded independently and decoding gives up to four different values for each pixel considered only.

4. Horizontal-Vertical Partitioning: The horizontal-vertical partitioning method the shape range not only a square also considered other rectangular because a range is divided into two rectangles as alternative into four squares. The frontier between the two rectangles is established in the most significant horizontal or vertical edge.

This method to disadvantages of quad tree partitioning it tries to find best division of a range into two new ranges by horizontal or vertical cut. The algorithm uses two formulas (*vn* and *hm*) that allow determining the direction and the position of the cut:

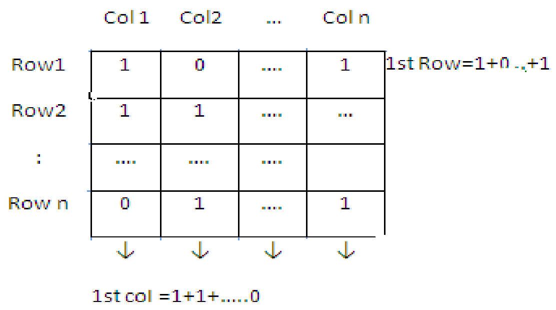
$$v_m = \frac{\min(m, \mathrm{width}(R_i) - 1 - m)}{\mathrm{width}(R_i)} .(\sum_{n=0}^{\mathrm{height}(R_4)-1} \tau_{m,n} - \sum_{n=0}^{\mathrm{height}(R_4)-1} \tau_{m+1,n}) \quad \ldots (5)$$

$$h_n = \frac{\min(n, \mathrm{height}(R_i) - 1 - n)}{\mathrm{height}(R_i)} .(\sum_{m=0}^{\mathrm{width}(R_4)-1} \tau_{m,n} - \sum_{m}^{\mathrm{width}(R_4)-1} \tau_{m,n+1}) \quad \ldots (6)$$

Where $\mathrm{width}(R_i)$ x $\mathrm{height}(R_i)$ is the dimension of range block $(R_i)$and$1 \le m < \mathrm{width}(R_i), 1 \le n < \mathrm{height}(R_i)$

Otherwise, the range block is partitioned vertically. This superiority is caused by better adaptively and larger range block sizes.

a) Generation of Feature Set: In this research work four different ways are used here for feature vector extraction of gray scale image. There for feature vector size for each method is different.



	Col 1	Col2	...	Col n
Row1	1	0	....	1
Row2	1	1	....	...
:	....	....	....	
Row n	0	1	....	1

Fig. 2 Calculation of Row & Column pixel Distribution

After calculating the column wise and row wise pixel distribution are getting one row and one column coefficients vector for bit pane. i.e *SB* Pr *ow* (*i* ,1) is the row wise pixel distribution and *SBPcol*(1, *j*) is the column wise pixel distribution for single bit plane. Calculate mean, Standard deviation of row vector and column vector using formula given in equation 3.7, 3.8, 3.9 and 3.10. Consider image size is 256 x 256.

$$\mu_{SBProw} = \frac{1}{256} \sum_{i=1}^{256} SB\,\mathrm{Pr}\,ow(i,1) \ldots (7)$$

$$\mu_{SBPcol} = \frac{1}{256} \sum_{j=1}^{256} SBP\,col(1, j) \quad \ldots (8)$$

$$\sigma_{SBProw} = \sqrt{\frac{1}{256} \sum_{i=1}^{256} (SBP_{row}(i,1) - \mu_{SBProw})^2} \ldots (9)$$

$$\sigma_{SBPcol} = \sqrt{\frac{1}{256} \sum_{j=1}^{256} (SBP_{col}(1, j) - \mu_{SBPcol})^2} \ldots (10)$$

Where μ, □ are the mean value, standard deviation value for the row and column vector of *SBP*(*i*, *j*) image. Now feature vector having size 4 is

$$FV = \{\mu_{SBProw,} \mu_{SBPcol,} \sigma_{SBProw,} \sigma_{SBPcol}\} \ldots (11)$$

5. Encoding: The fractal encoding can be done with 128 x 128 image in which each pixel can be one of 256 levels of gray called as picture range image. Then reduce by averaging the original image to 64 x 64 is known as new image domain image. After that, partitioned both images into blocks 4 x 4 pixels. To perform each block:

$$(D_{i,j}) = \alpha D_{i,j} + t_0 \quad \ldots (12)$$

Where $\alpha = [0,1]$ $\alpha \in \Re$ and $t_0 \in [-255, 255]$ $t_0 \in Z$

To find linear transformations of domain block to reach to the high-quality approximation of a given range block. Every area or domain block is transformed after which compared to every range block ($R_{k,\ l}$). The precise transformation on every domain block is remedy of a and t_o is observed minimizing.

$$\min \sum\nolimits_{m,n} (R_{k,l})_{m,n} - (\tau(D_{i,j}))_{m,n} \ldots (13)$$

With respect to a and *t*

$$\alpha = \frac{N_s^2 \sum_{m,n} (D_{i,j})_{m,n} (R_{k,l})_{m,n} - (\sum_{m,n} (D_{i,j})_{m,n}) \sum_{m,n} (R_{k,l})_{m,n})}{N_s^2 \sum_{m,n} ((D_{i,j})_{m,n})^2 - \sum_{m,n} ((D_{i,j})_{m,n})^2} \quad \ldots (14)$$

$$t_0 = \frac{(\sum_{m,n} (D_{i,j})_{m,n})^2 - \sum_{m,n} (R_{k,l})_{m,n})^2}{N_s^2 \sum_{m,n} ((D_{i,j})_{m,n})^2 - \sum_{m,n} ((D_{i,j})_{m,n})^2} \quad \ldots (15)$$

Where *m, n, Ns* = 2 or 4 (size of blocks

Every transformed area block G (Di,j) is compared to each range block Rk,l that allows you to find the nearest area or domain block to each range block. This evaluation is completed using the subsequent distortion measure.

$$d_{l_2}(\tau(D_{i,j}), R_{k,l}) = \sum_{m.n}((\tau(D_{i,j}) - (R_{k,l})_{m,n})^2 \quad (16)$$

Each distortion is stored and the minimal is selected. The transformed domain block that is located to be the satisfactory approximation for the current range block is assigned to that range block.

$$\tau(D_{i,j})\text{best} \Rightarrow R_{k.l} \qquad \ldots (17)$$

6. *Decoding:* The reconstruction process of the original image consists on the applications of the transformations describe in the fractal code book iteratively to some initial image W*init,* until the encoded image is retrieved back. The transformation over the whole initial image can be described as follows:

$$\Omega_1 = \eta(\Omega_{init})$$
$$\Omega_2 = \eta(\Omega_1)$$
$$\Omega_3 = \eta(\Omega_2)$$
$$\ldots.. = \ldots\ldots..$$
$$\Omega_n = \eta(\Omega_{n-1})$$

h can be expressed as two distinct transformations

$$\eta = \tau(\Omega)\Psi(\Omega) \ldots (18)$$

Where W to create a domain image

Y (W) represents the ensemble of the transformations defined by our mappings from the domain blocks in the domain image to the range blocks in the range image as recorded in the fractal.

IV. ALGORITHM

The entire procedure with the proposed features is presented as simple algorithms hereunder using MATLAB7.0

A. Encoding

Step 1: Consider the input image
Step 2: Compute the procedure_threshold ( )
Step 3: Select the dimensions size of the range blocks and domain blocks.
Step 4: Partition the input image into non-overlapped blocks of ranges of square sizes covering the entire image.
Step 5: Partition the image into non-overlapping domain blocks and rescale the domain blocks to the size of the range blocks.
Step 6: Construct the each domain block and examine the each range block with the entire domain blocks to find the best match.
Step 7: Keep the region of the domain, coefficients is called as encoded image is compressed.
Step 8: Maintain the identical for the all of the range blocks until it reaches the last range block.

B. Decoding

Step 1: Read in range block and reconstruct the block position, transform and size information.
Step 2: Use any blank opening image of identical size as original image.
Step 3: For each range block be valid stored transforms touching specific transform block.
Step 4: Overwrite range block pixel values with change block pixel values.
Step 5: Repeat until suitable image quality is reached.
Procedure _ threshold ( )
{
Step 1: Examine the image with even row and column.
Step 2: Convert gray scale values into matrix format.
Step 3: Apply sorting rule for an array through step 2.
Step 4: Find out middle gray scale values of lower range and upper range.
Step 5: Find the average value of middle gray scale values and take whole number in sorted array and additionally threshold value.
Step 6: Convert binary matrix by using threshold value.
Step 7: Repeat step 2 to step 6 for all images within the database.
Step 8: Return
}

V. PERFORMANCE PARAMETERS

The effectiveness of the compression is calculated with compression ratio (*CR*) is defined as number of bits of the original image (B_{org_img}) per one bit of the compressed image (B_{comp_img}),

$$CR = \frac{Bit_{org\_img}}{Bit_{comp\_img}} \ldots (19)$$

Mean square error is given by,

$$MSE = \frac{1}{M.N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[X(m,n) - \bar{X}(m,n)]^2 \quad \ldots (20)$$

The peak Signal to Noise ratio is calculated by:

$$PSNR = 10.\log_{10}\frac{{max_x}^2}{MSE} \ldots (21)$$

VI. EXPERIMENT AND RESULTS

In this work, fractal image encoding with thresholding and decoding for compression. The fractal image encoding with thresholding based compression reduces the encoding time as well as provides higher compression ratio.

Fig. 3 Original Image Range block size 4

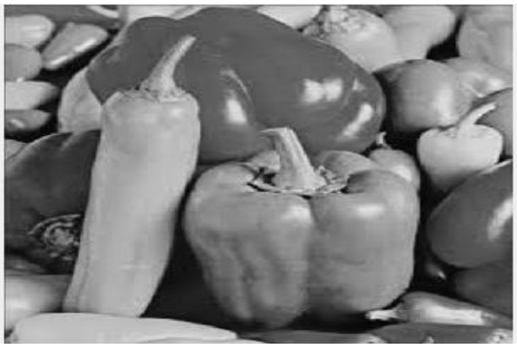


Fig. 4 Range block size 8

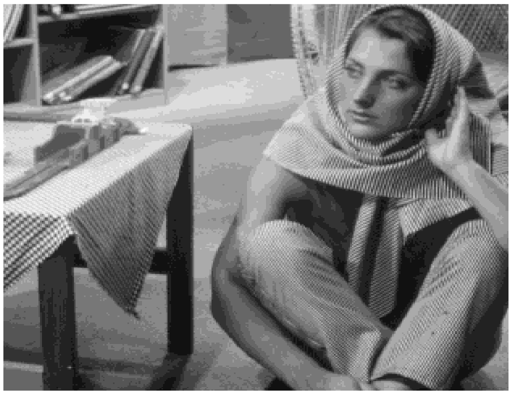


Fig. 5 Range block size16

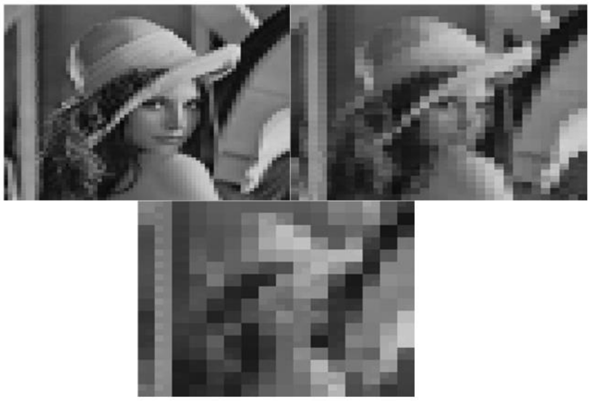


Fig. 6 Results Obtained Image

In order to evaluate the performance of the proposed system with the existing system the compressed file size and Compression Ratio (CR) are computed with the selective image sets. The obtained results of the image with the proposed model are tabulated in the following table I shows the performance is evaluated with the existing BTC, Wavelet and DCT.

From the below fig. 7 and fig. 8 shows the pictorial representation of the performance evaluation of file size compressed. By analysing the obtained results the proposed model produced the best results.

TABLE I COMPRESSED FILE SIZE

Block	BTC		Wavelet		DCT		Proposed BBF	
	Compressed File Size	CR	Compressed File Size	CR	Compressed File Size	CR	Compressed File Size	CR
2*2	196435	1.32	251559	1.15	257767	1.158	21762	17.43
4*4	141312	1.90	125576	2.32	216455	1.377	24561	10.17
8*8	90179	2.98	57376	5.08	219141	1.630	23737	9.52
16*16	53152	5.06	28681	10.17	218122	1.367	22563	10.58
32*32	32553	8.27	20963	13.91	213371	1.397	23891	15.04

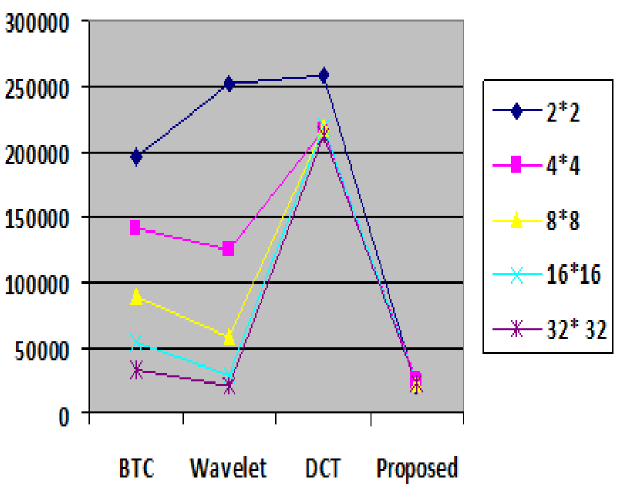


Fig. 7 Compressed File Size

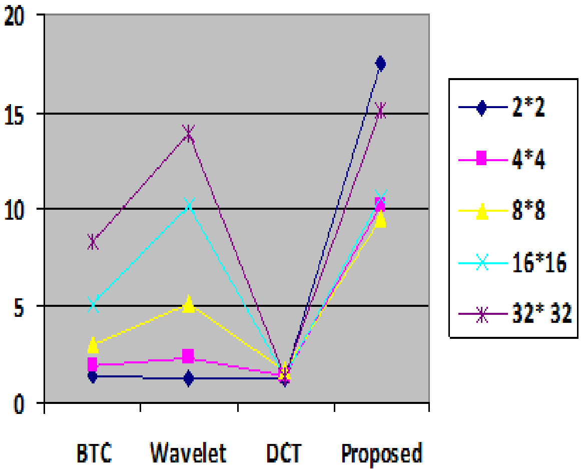


Fig. 8 Compressed Ratio

In order to evaluate the performance of the proposed system with the existing system the MSE are computed with the selective image sets. The obtained results of the image with the proposed model show the performance is evaluated with the existing BTC, Wavelet and DCT. From the below fig. 9 shows the pictorial representation of the performance evaluation of Mean Square Error. By analysing the obtained results the proposed model produced the best results.

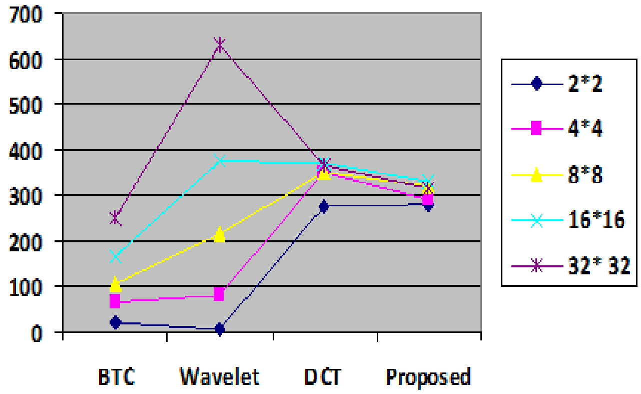


Fig. 9 MSE Comparison

In order to estimate the performance of the proposed system with the existing system the PSNR values are computed with the image. The obtained results of the image with the proposed model are tabulated in the following table II shows the performance is evaluated with the existing BTC, Wavelet and DCT.

TABLE II PSNR COMPARISON

Block	BTC	Wavelet	DCT	Proposed BBF
2*2	35.02	41.16	71.90	28.23
4*4	29.86	29.18	70.87	30.02
8*8	27.84	24.80	70.91	29.51
16*16	25.93	22.40	70.64	30.36
32* 32	24.17	20.15	70.70	31.29

From the below fig. 10 shows the pictorial representation of the performance evaluation of Peak-Signal to Noise Ratio. By analysing the obtained results the proposed model produced the best results.

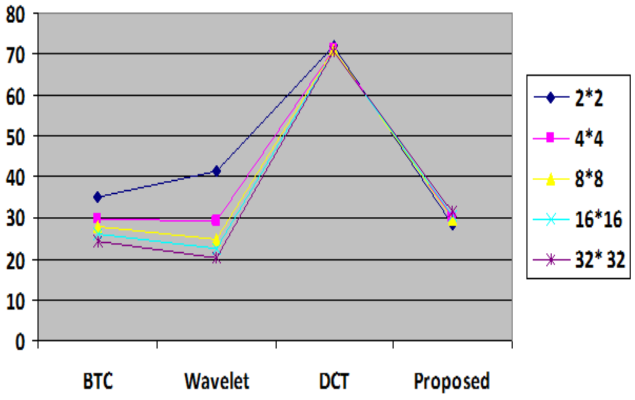


Fig. 10 PSNR Comparison

VII. CONCLUSION

In this paper, the image compression with block based Feature technique to generation of images features and the extraction of features with rectangular based has been presented. The experimental result proves the effectiveness of the block based methods provides good compression ratio, mean square error, Peak signal to noise ratio, encoding time and decoding time when compared to existing methods. The performances of block based feature method when compared to existing methods such as BTC, Wavelet and DCT are investigated independently. The block based feature compression technique produces better results when compared with existing methods. Moreover, the computational cost of the algorithm is very low also used for image compression.

REFERENCES

[1] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform", *IEEE Transactions on Computers*, Vol. 23, No. 1, pp.90-93, 1974.

[2] E.J. Delp and O.R. Mitchel, "Image Coding using Block Truncation Coding", *IEEE Transactions on Communications*, Vol. 27, pp. 1335-1342, 1979.

[3] Meftah M. Almrabet, Amer R. Zerek, and AllaouaChaouiAli A. Akash "Image compression using block truncation coding" *IJSTA*, Vol. 3, No. 2, pp. 1046-1053, 2009.

[4] Mehdi Salarian, Babak Mohamadinia, and JalilRasekhi, "A Modified No Search Algorithm for Fractal Image Compression", *International Conference on Image Processing, Computer Vision & Pattern Recognition, IPCV 2008*, Vol. 2, pp. 14-17, Las Vegas Nevada, 2008.

[5] Y. Chakrapani1 and K. Soundara Rajan, "Genetic Algorithm Applied To Fractal Image Compression ARPN" *Journal of Engineering and Applied Sciences*, Vol. 4, No.1, pp 53-58, 2009.

[6] Sachin Dhawan "A Review of Image Compression and Of Its Algorithms", *IJECT*, Vol. 2, No. 1, 2011.

[7] Sumathi Poobal, and G. Ravindran, "The Performance of Fractal Image Compression on Different Imaging Modalities Using Objective Quality Measures", *International Journal of Engineering Science and Technology*, Vol. 2, No.1, pp. 239-246, 2011.

[8] Anupam Garg "An Improved Algorithm of Fractal Image Compression", *International Journal of Computer Applications*, Vol. 34, No. 2, pp. 17-21 2011.

[9] Chetan Dudhagara and Kishor Atkotiya, "Experimental Study of Fractal Image Compression Algorithm", *International Journal of Computer Applications & Information Technology*, Vol. 1, No 2, pp. 18-24, 2012.

[10] D. Venkatasekhar and P. Aruna, "A Fast Fractal Image Compression Using Huffman Coding", *Asian Journal of Computer Science And Information Technology*, Vol. 2, No. 9, pp. 272– 275, 2012.

[11] Chandan Singh Rawat and Sukadev Meher "A Hybrid Image Compression Scheme Using of Information Technology DCT and Fractal Image Compression", *The International Arab Journal*, Vol. 10, No. 6, pp. 553-562, 2013.

[12] Jianji Wang and Nanning Zheng "A Novel Fractal Image Compression Scheme with Block Classification and Sorting Based on Pearson's Correlation Coefficient", *IEEE Transactions On Image Processing*, Vol. 22, No. 9, pp. 3690-3702, 2013.

[13] Jie He, and Hui Guo "Super-Sampling Method during Decoding For Fractal Image Compression" *Computer Modelling & New Technologies*, Vol. 18, No. 12, pp. 501-506, 2014.

[14] A. Krishnamoorthy "Fast Search Fractal Image Compression Using PSO Based Optimization Technique", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 14 No. 6, pp. 122-126, 2014.

[15] K. Raja Kumari and C. Nalini "Improvement of Image Quality Based On Fractal Image Compression". *Middle-East Journal of Scientific Research*, Vol. 20, No. 10, pp. 1213-1217, 2014.

[16] Preeti Banerjee, Deepak Kumar Xaxa, "Designing and Implementation of Efficient Fuzzy Logic Based Fractal Image Compression Technique". *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 3, pp. 3494-3499, 2014.

[17] S.V. Veenadevi and A.G. Ananth, "Fractal Image Compression of Satellite Color Imageries Using Variable Size of Range Block", *International Journal of Image Processing*, Vol. 8, No.1, pp. 1-8, 2014.

[18] A. R. Nadira Banu Kamal, "Iteration Free Fractal Image Compression for Color Images Using Vector Quantization, Genetic Algorithm and Simulated Annealing" *The Online Journal of Science and Technology*, Vol. 5, No.1, pp. 39-48, 2015.

[19] H. Miar Naimi and M. Salarian. "A Fast Fractal Image Compression Algorithm Using Predefined Values for Contrast Scaling." *World congress on Engineering and Computer Science*. Vol. 1, No. 4, pp. 1035-1039, 2015.

[20] K.S. Priyadarshini, and G.S. Sharvani "A Survey on Parallel Computing Of Image Compression Algorithms Jpeg and Fractal Image Compression" *Proceedings of the International Conference, "Computational Systems for Health & Sustainability*", pp. 17-18, 2015.

[21] Thai Nam Son, ThangManh Hoang, Nguyen Tien Dzung,Nguyen Minh Dung, and Pham Ngoc Thang "Fast Implementation of Fractal Image Compression" *IJCSNS* ,Vol. 2, No. 3, pp. 12-17, 2015.

[22] M. Salarian, and H. MiarNaimi "Modified Fast Fractal Image Compression Algorithm in Spatial Domain". *Computer Vision and Pattern Recognition* Vol. 2, No. 1, pp. 24-27, 2007.