Journal of Engineering and Applied Sciences 11 (7): 1549-1557, 2016
ISSN: 1816-949X

Study of Fractal Compression Technique for Saving Memory Space in Printing Technology

[1]Shimal Das and [2]Dibyendu Ghoshal
[1]Department of Computer Science and Engineering,
Tripura Institute of Technology, Narsingarh, India
[2]Department of Electronics and Communication Engineering,
National Institute of Technology, Agartala, India

Abstract: Image compression is becoming an indispensable technology to decrease memory space and to reduce transmission time, as printers begin to feature high resolution and color. All types of printer produce any image by reflection of light that is fundamentally a subtractive process. Printers most often employ Cyan, Magenta, Yellow (CMY) color space. Besides, they also use CMYK color space which incorporates black as an extra element in color vector. Fractal compression technique has been successfully employed in CMY and CMYK scheme to reduce the memory space of the printing system as well to enhance printing speed of the printers.

Key words: CMYK space, fractal compression, Iterated Function System (IFS), memory space saving, printing technology

INTRODUCTION

The clarity and color in the printed material become major issues; image data compression approaches are indispensable for reducing storage capacity and transfer time. Fractal geometry and its application in image as well as data processing have produced an impact in image processing (Mandelbrot, 1982, 1983; Barnsley, 1988, 1993; Barnsley and Jacquin, 1988; Jacquin, 1990, 1992, 1993; Belloulata and Konrad, 2002). Almost all data sets have got some redundant data which may be deleted without affecting the signal much. Fractal, being a lossy compression method, can well be applied for data compression. In digital imaging, red, blue and green are the primary color and cyan, magenta, yellow are secondary color as the latter three can be derived from RGB. In color pigments used in printing technology (may be printer, photo copier) cyan, magenta and yellow are called primary colors and red, blue and green are called secondary colors. When cyan pigments are suspended in liquid media it absorbs red components of the light and reflects the cyan, recombination of blue and green, etc. When cyan, magenta and yellow pigments are mixed in an equal amount in suspension, it produces a near black rather clay color not a perfect black. To overcome the problem, printing technology resorts to add extra black pigments which are to get black imprint. Thus the CMY color space is converted into CMYK scheme. This scheme has four components or vector planes thereby requiring four times more memory bits compared with ordinary gray scale images.

To reduce these huge amount of data set arising out of four component vectors, fractal compression method is applied in the present study. The 4×4 range and domain block segmentation and isosceles triangle methods have been applied to printed color images. It has been found that although a large number of a research work has been carried out (Brijmohan and Mneney, 2004; Cochran *et al*., 1996; Fisher, 1994; Zhao *et al*., 2000; Zhu *et al*., 2008; Lie *et al*., 2008a; Hurtgen and Stiller, 1993; Thomas and Deravi, 1995; Davoine *et al*., 1996; Polvere and Nappi, 2000 Hamzaoui *et al*., 2001) on colored images but no research report based on fractal compression for printing technology is available in online or published literature so far.

Literature review: Many researchers have proposed the color image compression process. In this study, a brief review of some important works from the existing literature is presented. There are several systems that implemented for image compression during previous years (Wallace, 1991). Wallace shows JPEG features a simple lossy

Corresponding Author: Shimal Das, Department of Computer Science and Engineering, Tripura Institute of Technology, Narsingarh, India

technique known as the Baseline method, a subset of the other DCT-based modes of operation. The Baseline method has been by far the most widely implemented JPEG method to date and is sufficient in its own right for a large number of applications. This study provides an overview of the JPEG standard and focuses in detail on the Baseline method.

Oka and Onishi (1992) they represent that a still-image coding, being standardized by JPEG (Joint Photographic Experts Group) will presumably be utilized for image communications or image data bases. The JPEG scheme can compress natural images efficiently but it is unsuitable for text or Computer Graphics (CG) images for degradation of restored images. This scheme, therefore, cannot be implemented for printers which require good image quality. Then they studied and implemented coding which are more suitable for printers than the JPEG scheme. Two criteria they considered to select a coding scheme for printers: no visible degradation of input printer images and capability of image edition. Especially in terms of criteria, a fixed-length coding was adopted; an arbitrary pixel data code can be easily read out of an image memory. Then they implemented an image coding scheme in their new sublimation full-color printer.

Katsuyuki and Matsushiro (1998) proposed an overview of the bi-level image data compression system, Quic-coder that they developed. This is a non-reciprocal bi-level image data compression system aiming at improving compression performance which is the first arithmetic coding system. Considering the relationship between high resolution and the visual characteristics of the printer, they designed the non-reciprocal characteristic such that the deterioration of images is hardly detected. Compared with the 2-level data compression system QM-Coder which is the international standard, data volume can be decreased 20-30%. Cosman and Zeger (1998) and Cosman *et al*. (1998) they present a method for ordering the wavelet coefficient information in a compressed bit stream that allows an image to be sequentially decoded with lower memory requirements than conventional wavelet decompression schemes. They also introduce a hybrid filtering scheme that uses different horizontal and vertical filters, each with different depths of wavelet decomposition. This reduces decoder memory requirements by reducing the instantaneous number of wavelet coefficients needed for inverse filtering.

Cosman *et al*. (1998), introduced the memory efficient image compression techniques for color digital printers. Particular constraints imposed by printers are incorporated into the algorithms presented. They present a method for ordering the wavelet coefficient information in a compressed bit stream to allow a decoded image to be sequentially decompressed, from the top of an image to the bottom, as the paper exits the printer. Moreover, they also use a hybrid filtering scheme that uses different horizontal and vertical filters, each with different depths of wavelet decomposition.

Brien introduce the JPEG Algorithm. The basis for the JPEG algorithm is the Discrete Cosine Transform (DCT) which extracts spatial frequency information from the spatial amplitude samples that examines each step in the compression sequence with special emphasis on the DCT. Sonal (2007) explain a study of various image compression techniques that presents the principal component analysis approach applied to image compression. PCA approach is implemented in two ways PCA statistical approach and PCA neural network approach. It also includes various benefits of using image compression techniques. Vehel *et al*. (2007) introduce ovecompressing JPEG images with evolution algorithms this evolutionary strategies are used in order to guide the modification of the coefficients towards a smoother image and the result was three compression ratios have been considered: The compressed images are obtained by using the quantization values in Table 1 multiplied by 5, 10 and 15 (Vehel *et al*., 2007).

Li *et al*. (2008b) used a detection algorithm for zero quantized DCT coefficients in JPEG show Experimental results show that the proposed algorithm can significantly reduce the redundant computations and speed up the image encoding. Moreover, it doesn't cause any performance degradation. Computational reduction also implies longer battery lifetime and energy economy for digital applications.

Rawat and Meher (2013) introduced performance evaluation of JPEG image compression using symbol reduction technique. In this study, a new technique has been proposed by combining the JPEG algorithm and symbol reduction huffman technique for achieving more compression ratio. The symbols reduction technique reduces the number of symbols by combining together to form a new symbol. As a result of this technique the number of Huffman code to be generated also reduced. The result shows that the performance of standard JPEG method can be improved by proposed method. This hybrid approach achieves about 20% more compression ratio than the Standard JPEG.

Ani and Awad (2013) proposed the basis for the JPEG algorithm is the Discrete Cosine Transform (DCT) which extracts spatial frequency information from the spatial amplitude samples. These frequency components are then quantized to eliminate the visual data from the image that is least perceptually apparent, thereby reducing the amount of information that must be stored. The redundant

Table 1: Comparison between PSNR (dB), execution time (sec) and compression ratio for 4×4 block segmentation (square partitioning) and isosceles triangle based segmentation, the results are based on the proposed algorithm

		4×4 block segmentation			Isosceles triangle based segmentation		
Images	CMYK	PSNR (dB)	Execution time (sec)	Compression ratio	PSNR (dB)	Execution time (sec)	Compression ratio
PINE	C	34.12	58	4.21	35.21	48	4.98
	M	33.39	54		34.33	48	
	Y	31.55	52		32.51	47	
	K	31.17	58		32.15	48	
Leopard	C	33.21	52	4.21	34.01	48	4.33
	M	33.04	58		33.22	47	
	Y	34.89	51		34.54	48	
	K	31.16	58		31.12	48	
Flower	C	33.11	56	4.19	34.20	47	4.45
	M	33.31	55		33.12	49	
	Y	31.25	56		34.34	49	
	K	31.11	54		32.10	48	
Penguins	C	33.10	53	4.26	35.11	47	4.76
	M	34.49	53		34.13	46	
	Y	33.51	54		34.15	47	
	K	33.94	55		33.12	47	

properties of the quantized frequency samples are exploited through quantization, run-length and Huffman coding to produce the compressed representation. Each of these steps is reversible to the extent that an acceptable approximation of the original space-amplitude samples can be reconstructed from the compressed form. This study examines each step in the compression and decompression. To saving the memory space and reduce encoding time, fractal compression technique for saving memory space in printing technology.

MATERIALS AND METHODS

Mathematical background of fractal based color image compression: The standard formula for fractal compression as well as affine transformation and intensity transformation are available in standard (Mandelbrot, 1982 1983; Barnsley and Jacquin, 1988; Jacquin, 1990, 1992, 1993; Belloulata and Konrad, 2002; Hurd *et al*., 1992; Brijmohan and Mneney, 2004; Cochran *et al*., 1996; Fisher, 1994; Zhao *et al*., 2000; Zhu *et al*., 2008; Li *et al*., 2008a; Hurtgen and Stiller, 1993; Thomas and Deravi, 1995; Davoine *et al*., 1996; Li *et al*., 2000; Zhu *et al*., 2010) Quadtree and isosceles triangle method of segmentation of range and domain blocks have been applied on printed documents.

Step 1: The RGB trichromatic concept may be extended for CMY and CMYK schemes. Let CMYK component of printed color image of size M,N are C_{ij}, M_{ij}, Y_{ij} and K_{ij}. Where i = 1, ..., N, j = 1, ..., 2, ..., N for (N, N) size. The difference between adjacent distance horizontal pixels or adjacent vertical pixel is 1. The quadric-chromaticity coefficient can be calculated as:

$$\left.\begin{aligned} X_{i,j} &= \frac{C_{i,j}}{C_{i,j}+M_{i,j}+Y_{i,j}+K_{i,j}} \\ Y_{i,j} &= \frac{M_{i,j}}{C_{i,j}+M_{i,j}+Y_{i,j}+K_{i,j}} \\ Z_{i,j} &= \frac{y_{i,j}}{C_{i,j}+M_{i,j}+Y_{i,j}+K_{i,j}} \\ W_{i,j} &= \frac{K_{i,j}}{C_{i,j}+M_{i,j}+Y_{i,j}+K_{i,j}} \end{aligned}\right\} \quad (1)$$

Step 2: The printed color image is divided into 4 equal square blocks. The average coefficient and variances can be obtained as:

$$\left.\begin{aligned} M_x &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y} X_{i,j} \\ M_y &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y} Y_{i,j} \\ M_z &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y} Z_{i,j} \\ M_w &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y} W_{i,j} \end{aligned}\right\} \quad (2)$$

$$\left.\begin{aligned} Var_x &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y}\left|X_{i,j}-M_x\right| \\ Var_Y &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y}\left|Y_{i,j}-M_Y\right| \\ Var_z &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y}\left|Z_{i,j}-M_z\right| \\ Var_w &= \frac{1}{m^2}\sum_{i=l_x}^{m+l_x}\sum_{j=l_y}^{m+l_y}\left|W_{i,j}-M_w\right| \end{aligned}\right\} \quad (3)$$

where, M_x, M_y, M_z and M_w represent average trichromatic coefficient of C, M, Y and K components, respectively. Var_x, Var_y, Var_z, Var_w denotes the variance of trichromatic coefficient X, Y, Z, W to the size of the current block. I_x, I_y indicates the coordinate of pixel of the top left corner in a block.

Step 3: The process of successive division of range block into another four quadrant sub-blocks which depend on the value of the variances encountered in the calculation the variance should be smaller than a threshold value where is predefined:

$$Var_x \leq = th, Var_y \leq = th, Var_z \leq = th, Var_w \leq = th \quad (4)$$

The position of the current block and average value of quadrichromatic coefficient are required to be stored. A new composite printed image S with intensely S (i, j) at location (i, j) is given by:

$$S_{i,j} = C_{i,j}X_{i,j} + M_{i,j}Y_{i,j} + Y_{i,j}Z_{i,j} + K_{i,j}W_{i,j} \quad (5)$$

The composite image is encoded during fractal compression.

Printed color image encoding and decoding technique based on fractal compression: Printed color image can be segmented either by successive division of image by four blocks ((Jacquin, 1990). The same fractal compression has been carried out on the basis of isosceles triangle segmentation (Zhao *et al.*, 2000). The whole fractal based encoding and decoding techniques are as follows:

Fractal image coding: A fractal is a geometric form which is based on self-similar irregular details in any object or image. The idea of fractal image coding is based on the assumption that a large amount of self-similarity is present in the image at the microscopic or block-image level. Thus, the image redundancies can be exploited by means of block-based self-affine transformations (Barnsley and Sloan, 1988; Jacquin, 1990). This is different from common transform coding approaches where a single invertible transform maps the image to a set of uncorrelated coefficients among which only the dominant ones are retained and further processed for storage and transmission. In the encoding phase, an image is partitioned into a number of disjoint blocks of size B×B called range blocks and a number of blocks of size 2B×2B called domain blocks. For each range block, the best matching domain block is searched in the domain pool (a collection of domain blocks) by performing a set of transformations on the blocks so that a given metric (e.g., root mean square) is minimized. Data compression is achieved by storing only the set of transformations, i.e., the fractal code, which contains the complete information required to reconstruct the image. The reconstruction (or approximation) of the original image is obtained by iterating the set of transformations on an arbitrary initial image.

Quad tree partitioning: The first practical block-based fractal coding scheme was developed by Jacquin (1990). The weakness of this approach is that some regions of the image may not be covered well due to the use of range blocks of fixed size. A quad tree-based fractal encoding scheme is an extension of the Jacquin's method and was used in this study. A quad tree partition is a representation of an image as a tree in which each node corresponds to a square of the image. Each node contains four sub nodes, corresponding to the four identical size quadrants of the square. The root of the tree is the original image. After some initial number of partitions is performed, the squares at the nodes (i.e., range blocks) are compared with domain blocks (which are twice the range size) in the domain pool. The size of the domain block is shrunk by pixel averaging to match the range size and the affine transformation (offset and scaling) of the pixel values is found by minimizing the Root Mean Squares (RMS) difference between the range pixel values and the transformed domain pixel values. Apart from offset and scaling, a domain block has eight possible isometric orientations (4 rotations and reflection with 4 rotations) to match a given range block. Thus, the domain pool can be thought of as being enlarged by including all rotations and reflections of each domain block. All the possible domain blocks are explored and compared with a range. If the depth of the quad tree is less than an assumed maximum depth and if the optimal RMS difference is larger than a threshold, the range block is further subdivided into four quadrants and the process is then repeated until the optimal RMS is less than the threshold. The set of transformations and domains are stored and the encoding process is completed.

Fractal printed color image encoding based on isosceles triangle segmentation: In this isosceles triangle segmentation the domain blocks and range blocks are divided into isosceles triangles rather than rectangles Fig. 1 show the range bocks, domain bock and range and domain bock segmentation approach (Zhao *et al.*, 2000). This segmentation method is in favor of approaching the diagonal edge and using the self-similar relationship of the image which helps reconstruct the edge information of original image. From Fig. 1a hypotenuse length of non-overlapped range blocks R_1, R_2,..., R_n is 4 and R_1, R_2, ..., R_n cover the whole image. Figure 1b and c are the

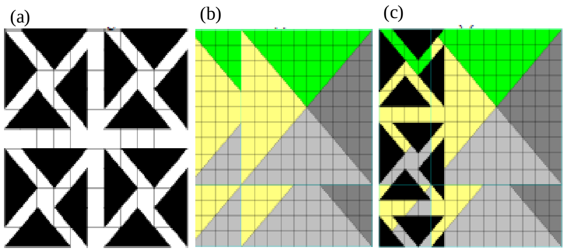

Fig. 1: a) Range block segmentation, b) domain block segmentation, c) segmentation scheme of range blocks and domain blocks (Zhao *et al*., 2000)

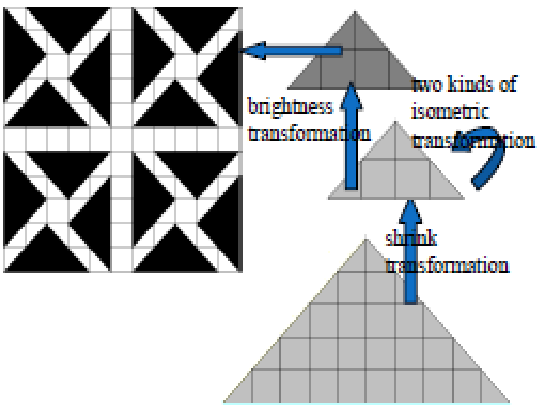

Fig. 2: Matching process between domain and range blocks (Zhao *et al*., 2000)

overlapped domain block segmentations with hypotenuse length of 10. Step length of domain blocks is L. In other words, number of pixels between two adjacent domain blocks is L. The pixel number of domain blocks is four times more than that of range blocks. The step length of domain block would influence the efficiency and image quality of fractal coding directly. The shorter the step length, the more matching times between domain blocks and range blocks is needed. However, over-length step may result in mismatching between blocks leading to the reducing of the image quality.

For finding the best matching domain block, every domain block needs to do shrink transformation, isometric transformation, brightness transformation and brightness excursion as shown in Fig. 2. The most similar domain block of current range block is found after comparing range blocks with transformed domain blocks. Then record the corresponding coefficient of affine transformation. It is the coefficients which construct fractal code of current range block.

In third used fractal based image encoding for getting the higher compression ratio and PSNR and decreasing the encoding time based on self- similarity. Figurre 3 shows the encoding process. Proposed algorithms encoding steps as following:

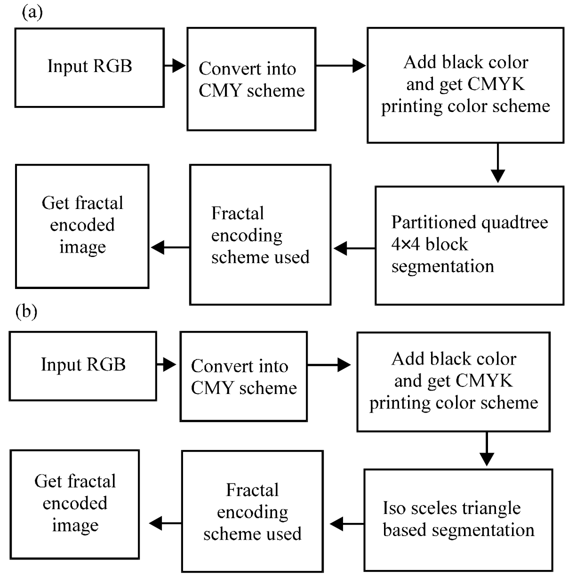

Fig. 3: a) Shows the encoding process based on quadtree partitioned, b) shows the encoding process based on isosceles triangle segmentation

Step 1: Divide initial image into different R blocks with non-overlapping, each R block is 8×8 size and set label for each R block, classify R block based on isosceles triangle segmentation rather than rectangles.

Step 2: Divide image into different D blocks allowing some overlapping, the size is four times as R block and set label for each D block, classify D block based on isosceles triangle segmentation rather than rectangles.

Step 3: Shrink D block to R block's size.

Step 4: Calculate D block's variance and for D blocks in the same class, sort them according to their variance.

Step 5: Choose one R block; calculate its variance and matching threshold.

Step 6: In the search space where R block located in, find a D block which has the closest variance to current R block.

Step 7: In the searching window, after affine transformation and gray migration, find the best-match D block.

Step 8: Calculate the matching error between best-match D block and R block, if the error is larger than threshold 'bias', divide R block into four sub-blocks, add them to the corresponding search space according to their labels, and slide certain steps, divide some new D blocks, add them to corresponding search space according to their labels too. If the error is smaller than threshold bias, storage affine transformation parameters; now, calculate the variance of range and domain blocks by the equation below. The variance of block I is defined as:

$$\mathrm{Var}(I) = \sqrt{\left\{\frac{1}{n\times n}\left(\sum_{i=1}^{n\times n} X_i^2\right) - \left(\frac{1}{n\times n}\left(\sum_{i=1}^{n\times n} X_{i=1}\right)\right)^2\right\}} \quad (6)$$

Where:

n = The size of the block and

X_i = The pixel value of the range blocks

Step 9: For all R blocks, repeat step 5-9.

Step 10: Write out the compressed data in the form of a local IFS code.

Step 11: Apply a fractal compression algorithm to obtain a compressed IFS code.

From Fig. 3a, b shows the overall encoding process based on quadtree partitioned and isosceles triangle segmentation.

RESULTS AND DISCUSSION

Four printed color photographs have been considered for the present study. The color content distribution should be uniform or by self similar in nature. A simple stroke by a paint brush on a screen could change the self similarity to a great extent. Nowadays a huge volume of work is going on in printing technology and serves the purpose. To store the content in soft copy form in memory space by preserving perfect black color, especially on pigmented image has posed a real problem. The memory space required to store a pigments colored image is four times more than that required to store gray level image, when the resolution and quantization bits remain the same. The fading up of black color is yet another problem and warrants a solution by storing them in appropriate manner. Fractal compression, although a lossy one, can provide compression ratio at least >4 in the decoded retrieved photography. The processing time depends on the number of range blocks and domain blocks as well as the predefined threshold value.

The significance of the present work lies in the fact in introducing the concept of quadric chromatic coefficient in a composite color pigment suspension or a streak of color of a suitable screen. The concept was inevitable and introduced by keeping a similarity with tri-chromatic coefficient. The PSNR calculation is carried out in a routine manner to gauze the efficiency of the applied algorithm and it has been found that the proposed algorithm involving quadric chromatic coefficient and corresponding variances bear sufficient relevance in the context of partition iterative function system. The processing time is also found to be bit higher which have been obtained in case of Jacquin (1990) and Zhao *et al*. (2000). The experimental results have established the importance of fractal compression in CMYK color pigmented images.

Figure 4 shows the original and reconstructed images of Pine, Leopard image, flower and penguins after decoding and Table 1 shows the comparison between PSNR (dB), execution time (in sec) and compression ratio for 4×4 block segmentation (square partitioning) and

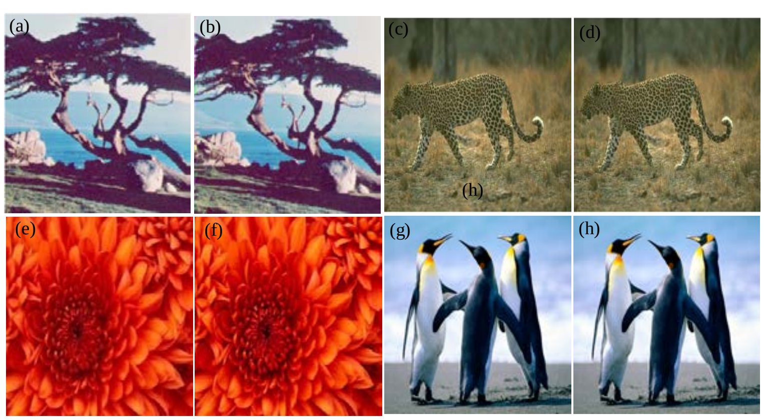

Fig. 4: a) Origional image of pine, b) reconstructed image of pine, c) origional Leopard image, d) reconstructed Leopard image, e) origional image of flower, f) reconstructed image of flower, g) origional image of Penguins and h) reconstructed image of Penguins

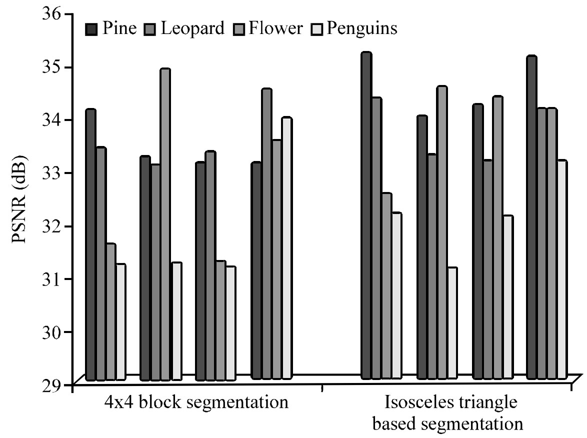

Fig. 5: The comparison chart of PSNR (dB) of four images as for 4x4 block segmentation and isosceles triangle based segmentation

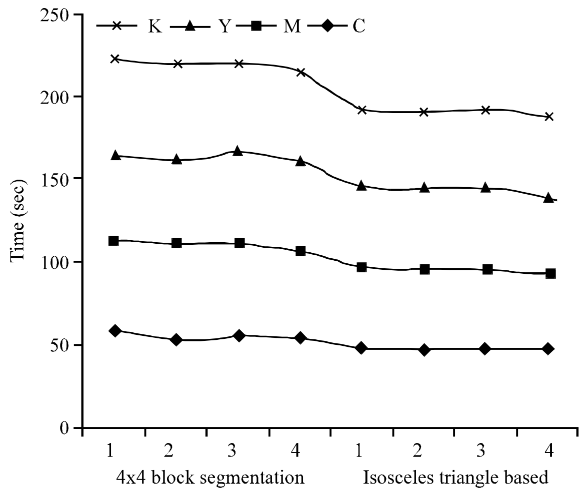

Fig. 6: The comparison chart of TIME (sec) of four images as; 1) Pine, 2) Leopard image, 3) Flower and 4) Penguins

isosceles triangle based segmentation. Figure 5 shows the comparison chart of PSNR (dB) of four images as Pine, Leopard image, flower and penguins for 4×4 block segmentation and Isosceles triangle based segmentation approach. The value of PSNR slightly increase in Isosceles triangle based segmentation approach compared than 4×4 block segmentation approach.

Figure 6 shows the comparison chart of time (sec) of four images as Pine, Leopard image, flower and penguins for 4×4 block segmentation and Isosceles triangle based segmentation approach.

In Fig. 7 shows the comparison chart of compression ratio of four images as Pine, Leopard image, flower and penguins for 4×4 block segmentation and isosceles triangle based segmentation approach.

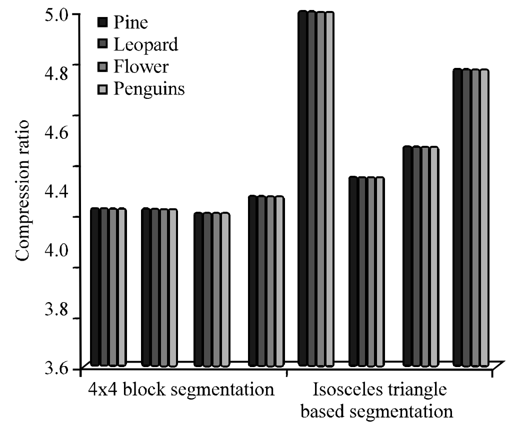

Fig. 7: The comparison chart of compression ratio of four images as Pine, Leopard image, flower and Penguins

CONCLUSION

To realize true black color in printing technology always poses vast problem. To solve the issue, the compression and segmentation of four color images has been thoroughly dealt with. The black color components need maximum processing time where the compression ratio becomes constant for all three color pigment components. The concepts of quadric-chromatic coefficient may lead to further research in printing technology in near future.

REFERENCES

Ani, M.S.A. and F.H. Awad, 2013. The JPEG image compression algorithm. Int. J. Adv. Eng. Technol., 6: 1055-1062.

Barnsley, M., 1993. Fractals Everywhere. 2nd Edn., Academic Press, San Diego, CA., USA., ISBN-10: 0120790610, pp: 550.

Barnsley, M.F. and A.D. Sloan, 1988. A better way to compress images. Byte, 13: 215-223.

Barnsley, M.F. and A.E. Jacquin, 1988. Application of recurrent iterated function systems to images. Proceedings of the SPIE Conference on Visual Communications and Image Processing' 88: Third in a Series, October 25, 1988, International Society for Optics and Photonics, Cambridge, Massachusetts, pp: 122-131.

Barnsley, M.F., 1988. Fractal Everywhere. Academic Press, New York, USA.

Belloulata, K. and J. Konrad, 2002. Fractal image compression with region-based functionality. IEEE Transact. Image Process., 11: 351-362.

Brijmohan, Y. and S.H. Mneney, 2004. Low bit-rate video coding using fractal compression of wavelet subtrees. Proceedings of the 7th Africon Conference on Africa (Africon04), Vol. 1, September 15-17, 2004, IEEE, Durban, South Africa, ISBN:0-7803-8605-1, pp: 39-44.

Cochran, W.O., J.C. Hart and P.J. Flynn, 1996. Fractal volume compression. IEEE. Trans. Visualization Comput. Graphics, 2: 313-322.

Cosman, P. and K. Zeger, 1998. Memory constrained wavelet based image coding. IEEE. Signal Process. Lett., 5: 21-23.

Cosman, P., T. Frajka and K. Zeger, 1998. Image compression for memory constrained printers. Proceedings of the International Conference on Image Processing (ICIP98), October 7, 1998, IEEE, La Jolla, San Diego, California, ISBN:0-8186-8821-1, pp: 109-113.

Davoine, F., M. Antonini, J.M. Chassery and M. Barlaud, 1996. Fractal image compression based on delaunay triangulation and vector quantization. IEEE. Trans. Image Process., 5: 338-346.

Fisher, Y., 1994. Fractal image compression. Fractals, 2: 347-361.

Hamzaoui, R., D. Saupe and M. Hiller, 2001. Distortion minimization with fast local search for fractal image compression. J. Visual Commun. Image Represent., 12: 450-468.

Hurtgen, B. and C. Stiller, 1993. Fast hierarchical codebook search for fractal coding of still images. Proceedings of the Conference on Video Communications and PACS for Medical Applications Berlin-DL Tentative, October 29, 1993, International Society for Optics and Photonics, Berlin, Germany, pp: 397-408.

Jacquin, A.E., 1992. Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Process., 1: 18-30.

Jacquin, A.E., 1990. A novel fractal block-coding technique for digital images. Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing ICASSP-90, April 3-6, 1990, IEEE, Atlanta, Georgia, USA, ISBN: 1520-6149, pp: 2225-2228.

Jacquin, A.E., 1993. Fractal image coding: A review. Proc. IEEE., 81: 1451-1465.

Katsuyuki, I. and N. Matsushiro, 1998. Image data compression techniques for printers. OKI. Techn. Rev., 64: 73-76.

Li, J., J. Takala, M. Gabbouj and H. Chen, 2008a. A detection algorithm for zero-quantized DCT coefficients in JPEG. Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, March 31-April 4, 2008, IEEE, Changchun, China, ISBN:978-1-4244-1483-3, pp: 1189-1192.

Li, J., D. Yuan, Q. Xie and C. Zhang, 2008b. Fractal image compression by ant colony algorithm. Proceedings of the 9th International Conference for Young Computer Scientists ICYCS 2008, November 18-21, 2008, IEEE, Jinan, China, ISBN: 978-0-7695-3398-8, pp: 1890-1894.

Li, Z.H., L. Zhao and N.Y. Soma, 2000. Fractal color image compression. Proceedings 13th Brazilian Symposium on Computer Graphics and Image Processing, October 17-20, 2000, Gramado, pp: 185-192.

Mandelbrot, B.B., 1982. The Fractal Geometry of Nature. 1st Edn., W.H. Freeman, San Francisco, CA., ISBN: 0716711869.

Mandelbrot, B.B., 1983. The Fractal Geometry of Nature. Macmillan, New York.

Oka, K. and M. Onishi, 1992. Implementation of image compression for printers. Proceedings of the SPIE-IS&T 1992 Symposium on Electronic Imaging: Science and Technology, May 1, 1992, International Society for Optics and Photonics, San Jose, California, pp: 450-454.

Polvere, M. and M. Nappi, 2000. Speed-up in fractal image coding: Comparison of methods. IEEE. Trans. Image Process., 9: 1002-1009.

Rawat, C. and S. Meher, 2013. A hybrid image compression scheme using DCT and fractal image compression. Int. Arab. J. Inf. Technol., 10: 553-562.

Sonal, D.K., 2007. A study of various image compression techniques. Coit Cleaners Company, Guru Jhambheswar University of Science and Technology, Rimt-IET, Hisar, India.

Thomas, L. and F. Deravi, 1995. Region-based fractal image compression using Heuristic search. IEEE Trans. Image Process., 4: 823-838.

Vehel, J.L., F. Mendivil and E. Lutton, 2007. Overcompressing JPEG Images with Evolution Algorithms. In: Applications of Evolutionary Computing, Giacobini, M. (Ed.). Springer, Berlin, Germany, ISBN:978-3-540-71804-8, pp: 383-390.

Wallace, G.K., 1991. The JPEG still picture compression standard. Commun. ACM, 34: 30-44.

Zhao, Y., Z. Zhu and H. Yu, 2010. Fractal color image coding based on isosceles triangle segmentation. Proceedings of the 2010 International Workshop on Chaos-Fractals Theories and Applications (IWCFTA), October 29-31, 2010, IEEE, Shenyang, China, ISBN: 978-1-4244-8815-5, pp: 486-490.

Zhu, S., L. Yu and K. Belloulata, 2008. An improved fractal image coding algorithm based on adaptive threshold for quadtree partition. Proceedings of the Seventh International Symposium on Instrumentation and Control Technology, October 13, 2008, International Society for Optics and Photonics, Bellingham, Washington, pp: 71290O-71290O.

Zhu, Z.L., Y.L. Zhao and H. YU, 2010. Efficient fractal image compression based on pixels distribution and triangular segmentation. J. Comput. Appl., 2: 337-340.