Proc. of the Intl. Conf on Advances in Computer and Information Technology - ACIT 2012
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-981-07-3161-08

Iteration free Fractal Compression using Simulated Annealing for Still Colour Images

Dr A R Nadira Banu Kamal
Department Of Computer Science,
TBAK College for Women, Kilakarai,
Ramanathapuram, India.

Abstract--**An iteration-free fractal coding for still colour image compression is proposed using simulated annealing. As the proposed method uses the synthetic code book, it reduces the coding process time and minimizes intensive computations. The proposed technique utilizes the SA, which greatly decreases the search complexity of matching between range block and domain block. Parameters such as image quality, compression ratio and coding time are analyzed. It is observed that the proposed method achieves excellent performance in image quality with very low bit rate.**

Key words-- **Fractal image compression; simulated annealing; synthetic code book; iteration free; colour images**

I. INTRODUCTION

Fractal Image Compression is described as a self-vector Quantization, where the image blocks are encoded applying a simple transformation to one of the blocks previously encoded. Transformations frequently used are combinations of scaling, reflection and rotation of another block. In the fractal coding schemes, an image is partitioned into non overlapping range blocks. The larger domain blocks D are selected from the same image which can overlap. A color image is encoded by mapping the domain block D to the range block R with the contractive affine transformation given by Eq. (1)

$$\hat{R} = i\{\alpha.D + \mu_R - \alpha.\mu_D\} = i\{\alpha.(D - \mu_D) + \mu_R\} \quad (1)$$

The parameters (called the *fractal code*) describing the contractive affine transformation, which has the minimum matching error between the original range block R and the coded range block $\hat{R}$, are transmitted or stored. The fractal code consists of the contrast scaling α, the block mean (the average pixel value of the range block) μ_R, isometry *i*, and the position P_D of the best-match domain block in the domain pool.

The fractal image compression problem puts forward three major requirements: speeding up the compression algorithm, improving image quality and increasing compression ratio [3, 6]. An iteration-free fractal image coding using the technique Simulated Annealing (SA) is proposed for lossy compression in this research work to improve decoded image quality, compression ratio and to reduce the coding time. Usage of synthetic codebook for encoding using fractal does not require iteration at decoding and the coding error is determined immediately at the encoder [7]. Hence there is a reduction in decoding time. Very few literatures are available on iteration-free fractal coding. Simulated Annealing is based on annealing in metallurgy to solve optimization problems [8]. Instead of searching one point at a time, they use multiple search points. Thus, they claim significant advantage of large reduction in search space and time. Optimal fractal coding is an NP-hard combinatorial optimization problem [2, 4]. So this technique is applied in this research work for fractal image compression.

An iteration-free fractal coding scheme with vector quantization was proposed in [1]. An improvement for reduction in time is suggested in the proposed method using SA, as SA can be applied to virtually any problem that has a large search space. It attempts to find near-optimal solutions without going through an exhaustive search mechanism. Hence it is proposed to use SA for finding the better match of the domain block to the range block without going through exhaustive search in the iteration-free fractal coding instead of the existing method using VQ.

In this research paper section II and III describes the architecture and the algorithm of the proposed method. Section VI explains how it is implemented followed by results and discussions in section V. Conclusion and its applications are given in section VI.

II. Architecture of the Proposed Technique

In the proposed method a synthetic codebook is created as the domain pool using the mean image, whose pixel values are the block means of all the range blocks. This code book is used as the domain pool for SA technique. The architecture of the proposed method is described in Figure 1.

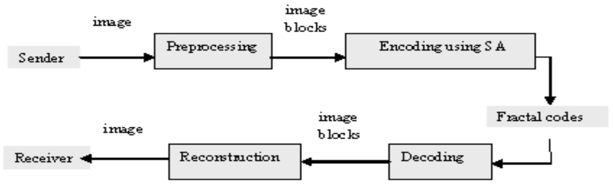


Figure 1. Architecture Of The Proposed Iteration-Free Fractal Image Coding Method

The sender sends the color image for compression. In the preprocessing stage, the input MxN image under coding is divided into non-overlapping square blocks of BxB pixels called the range blocks. Then the mean and variance of each range blocks are determined. For each range block the red, green and blue component's mean and variance are computed and then concatenated. After the mean of all the range blocks are obtained, a mean image of size M/B x N/B with each pixel corresponding to the block mean is generated. The mean image must be larger than the size of the range block i.e. M/B x N/B > B x B. The maximum size of B is limited to 8 in order to produce a good quality of the decoded image. The higher the resolution of the input image (MxN) more blocks can be generated for the domain pool which helps to find a good mapping between the domain and range blocks. The initial domain pool with blocks of the same size as the range is generated using the mean image. In the encoder if the variance

$$V\{R\} = \frac{1}{B^2} \sum_{0 \le i,j < B} (r_{i,j} - \mu_R)^2 \tag{2}$$

of the range block is smaller than the threshold value *E*, the range block is coded by the mean, or else the range block will be coded by the contractive affine transformation [9]. The aim of the proposed scheme is to find the domain block for each image range block and the transformation parameters that minimize the distortion between the image block and the transformed domain block in a minimized time. This process of finding the best domain block makes use of the techniques SA.

In the decoder, shown in Figure 2 the mean information of each range block is extracted from the fractal codes. Using this information the mean image is constructed. This mean image is partitioned into blocks of the same size as the input image. This forms the domain pool for SA search methods. The decompressed image is constructed block by block by applying the transformation parameters to the selected domain block from the domain pool as per the code.

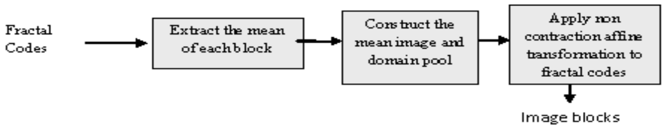

Fractal Codes
Extract the mean of each block
Construct the mean image and domain pool
Apply non contraction affine transformation to fractal codes
Image blocks

Figure 2. Proposed Decoder

A. *Encoder for the Proposed Iteration-free Fractal Image coding using SA*

The random numbers required for SA is globally generated using Knuth algorithm for random numbers. The possible system configurations, generator of random changes in configuration, means of evaluating the problem function and annealing schedule are finalized only by trial that results in good compression ratio and PSNR. The SA method is applied to search the best domain block with the required transformation that match the range block. The architecture of the encoder using SA is described in Figure.3.

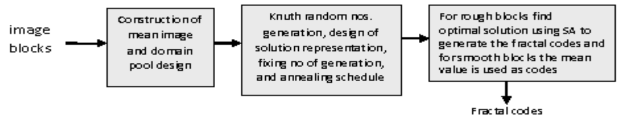


image blocks
Construction of mean image and domain pool design
Knuth random nos. generation, design of solution representation, fixing no of generation, and annealing schedule
For rough blocks find optimal solution using SA to generate the fractal codes and for smooth blocks the mean value is used as codes
Fractal codes

Figure 3. Proposed Encoder Using SA

The number of possible domain blocks to be searched is (M/B^2) x (N/B^2), the number of isometry transformations to be searched for each domain block is eight and the contrast scaling parameter is four for each RGB color components. Thus, the space to be searched consists of N1 elements for each RGB color components. N1 = 8 x 4 x (M/B^2) x (N/B^2). Let the space to be searched for each RGB color components be represented by P where

$$P = \{1,2,\ldots,8\} \times \{1,2,3,4\} \times \{1,2,\ldots,(M/B^2) \times (N/B^2)\} \tag{3}$$

Binary strings are introduced to represent the elements of P

[5, 10]. The set of 2^n binary strings, each of length n for each RGB color components, are constructed in such a way that the set exhausts the whole parametric space. The value for n depends on the values of M, N and B. The fitness function between the given range block and the obtained range block is taken to be the MSE given in Equ. 4. In order to apply the SA method to a specific problem, one must specify the state space, the neighbor selection method (which enumerates the candidates for the next state **s'**), the probability transition function, and the annealing schedule. These choices can have a significant impact on the method's effectiveness. At each step, the SA heuristic considers some neighbors of the current state **s**, and probabilistically decides between moving the system to state **s'** or staying back in state **s**. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Boltzmann's Probability is used in the proposed method. In the present problem the neighbor of a state is any domain block from the domain pool. Another essential feature of the SA method is that the control parameter is gradually reduced as the simulation proceeds. Initially, T is set to a high value (or infinity), and it is decreased at each step according to some annealing schedule — which may be specified by the user, but must end with T=0 towards the end of the allotted time budget. Cooling schedule $T_i = T_0 - i\,(T_0 - T_N) / N$ is used in the proposed method. T_i is the temperature for cycle *i*, where *i* increases from 0 to N. The initial and final temperatures, T_0 and T_N respectively, are determined by the user, as is N. In the proposed method the value of N is chosen to be 40. The probability of making the transition to the new state **s'** is a function P(δE, T) of the energy difference δE = E(s') - E(s) between the two states, and of a global time-varying parameter T called the control parameter. In the proposed method T is made to vary from 1 to 0. The fitness function between the given range block and the obtained range block is taken to be the Mean Square Error (MSE) as given in (4)

$$MSE(R, \hat{R}) = \frac{1}{B^2} \sum_{0,i,j \le B} \left(r_{i,j} - \hat{r}_{i,j} \right)^2 \tag{4}$$

III. Algorithms for Encoding and Decoding of the Proposed Iteration-free Fractal Image Coding

Proposed Encoder

The encoding procedure can be summarized in the following steps.

Step1: The mean μ_R and variance V_R of each range block R(i,j) is determined. A mean image of size M/B x N/B with each pixel corresponding to the block mean is generated. The mean image must be larger than the size of the range block i.e. M/B x N/B > BxB.

Step2: The mean image is divided into blocks of the same size as the range block (B x B pixels) to form the domain pool.

Step3: If the variance of the range block is smaller than the threshold value *E*, then the range block (smooth block) is coded by the mean, otherwise, the range block (rough block) will be coded by the fractal code f (i, α, μ_R, P_D) using SA technique. Here i represent the isometry transformations, α the contrast scaling, μ_R the mean value of the range block and P_D the domain block number in the domain pool.

$$\hat{R}(i,j) \begin{array}{l} = \mu_R \\ = f(i, \alpha, \mu_R, P_D) \end{array} \left| \begin{array}{l} \text{if } V_R = E \\ \text{if } V_R > E \end{array} \right. \qquad (5)$$

Thus the input from the sender is the image and the output is the fractal codes.

A. *Algorithm of the Encoder for the Proposed Iteration-free Fractal Image coding using SA*

The algorithm of the proposed iteration-free fractal image coding using SA is given as follows:

Step1: Start by generating an initial solution s (randomly) and by initializing the control parameter T.

Step2: Then the following is repeated until the termination condition is satisfied:

Step3: A solution s' is randomly sampled and it is accepted as new current solution depending on f(s), f(s') and T.

Step4: s' replaces s if f(s') < f(s) or, in case f(s') >= f(s), with a probability following the Boltzmann distribution exp(-(f(s') - f(s))/T).

B.*Algorithm of the Decoder*

The decoding process is as follows:

Step1: Extract the mean information of each range block from the fractal codes and construct the mean image.

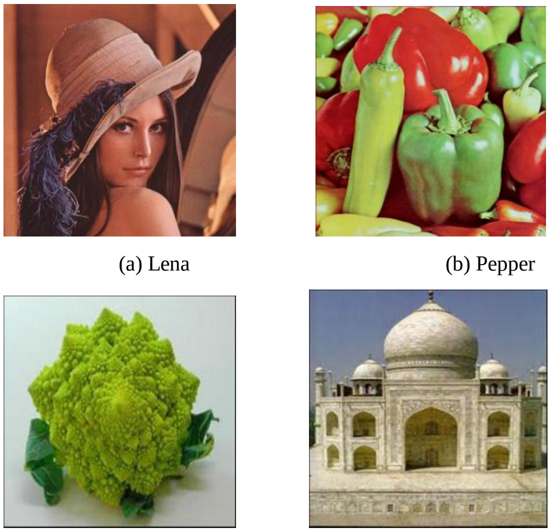
Step2: The domain pool is obtained by partitioning the mean image using the same size as the range block for SA.

Step3: For smooth blocks, the decompressed image blocks are obtained by the mean value and for rough blocks apply contractive affine transformation using the fractal codes.

The outputs of the decoder are image blocks that are combined to form the decoded image at the receiver end. Thus the receiver gets the fractal codes as input and the decompressed image as output.

IV. Implementation

These algorithms were implemented using the software Matlab 7.12 on the Intel (R) Core[TM]2 E7500 system with 2.93 GHz and 1.96 GB of RAM. For implementation of these algorithms, four 512 x 512 benchmark color images of Lena, Pepper, Cauliflower and Tal Mahal [shown in Figure 4 (a) to (d)] with twenty four-bit RGB color resolution were used.



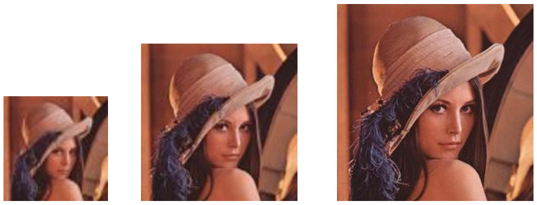
(a) Lena (b) Pepper

(c) Cauliflower (d)Taj Mahal

Figure 4. Original (512 X 512, 24 Bit/Pixel) Images.

In the simulation, the images were partitioned into range blocks with the single size, either 8x8 or 4x4 or 2x2. The maximum block size is set to 8x8 because for a range block size greater than 8x8 the determination of the proper domain block was difficult and the quality of the image reconstructed was poor. The threshold value for the variance of range blocks was chosen by trial and error basis to be of size 20 for block size 8x8, 10 for 4x4 and 5 for 2x2 that results in good compression ratio and PSNR. The number of blocks in the mean image is the size of the domain pool.

The range block with a single size (8x8, 4x4 & 2x2) was considered for simulation. Here the total number of range blocks for the block size 4x4 for each RGB color component was n = 16384 and total number of domain blocks (m) to search for each RGB color component were (128 / 4) x (128 / 4) = 32 x 32. Thus, the cardinality (N1) of the search spaces for each RGB color component of this case was 8 x 4 x 1024.The string length n for each RGB color component was taken to be 15 (3 + 2 + 10). Hence, the search space reduction ratio was approximately 14.



(a) 64x64 (b) 128x128 (c)256x256

Figure 5. Mean Image of Lena for block size 8x8, 4x4 and 2x2

The coding performance of the proposed method using the parameters like decoded image quality, bit rate and encoding time was determined. For the image partitioned by 8x8, 4x4 and 2x2 range blocks, the 64x64, 128x128 and 256 x 256 mean image for Lena was obtained and shown in Figure 5 (a), (b) and (c) respectively.

V. Results and Discussions

The range blocks were classified before coding. Range blocks were grouped into two sets according to the variability of the pixel values in these blocks. If the variability of a block was low, i.e., if the variance of the pixel values in the block was below a fixed value, called the threshold, the block is called smooth type range block. Otherwise, it is called a rough type range block. The purpose of choosing this block classification was for two reasons. One is to get higher compression ratio, and the other is to reduce the coding time. The threshold value that separates the range blocks into two types was chosen as stated earlier. After classification, SA-based coding was adopted for the rough type range blocks only. All the pixel values in a smooth type range block were replaced by the mean of its pixel values. This scheme is a time-saving one provided; the number of smooth type range blocks is significant. The storage requirements for the proposed method can be calculated from the number of smooth and rough blocks multiplied by the number of bits required to store the values. Table I & II (a) gives the classification of blocks, coding time, PSNR and bit rate using different types of encoding using the proposed technique on the color images chosen for simulation. From the results tabulated in Table 1, it is observed that for the images which have the number of smooth blocks significantly high has a high compression ratio.

Image	Range Block size	RMS	PSNR
Lena	2 * 2	4.32	35.41
	4 * 4	6.64	31.68
	8 * 8	9.64	28.36
Pepper	2 * 2	3.92	36.24
	4 * 4	6.25	32.20
	8 * 8	10.33	27.84
Cauliflower	2 * 2	3.42	37.44
	4 * 4	9.78	28.31
	8 * 8	14.52	24.88
Tajmahal	2 * 2	2.57	39.90
	4 * 4	6.16	32.32
	8 * 8	12.02	26.52

Figure 6 (a), (b) and (c) shows the decompressed Lena image using the proposed method for a single level partition of size 8x8, 4x4 and 2x2. The RMS of the decoded image partitioned by the 8x8 block size is higher than that partitioned by the 4x4 and 2x2 block size since a smaller block size leads to a smaller matching error for the affine transformation. However, the bit rate increases significantly because the number of the 2x2 range blocks is four times the number of the 4x4 range blocks and number of the 4x4 range blocks is four times the number of the 8x8 range blocks. The decompressed image of Pepper, Cauliflower and Taj Mahal for the single block partition of sizes 8x8, 4x4 and 2x2 using SA are shown in Figure 7, 8, 9.

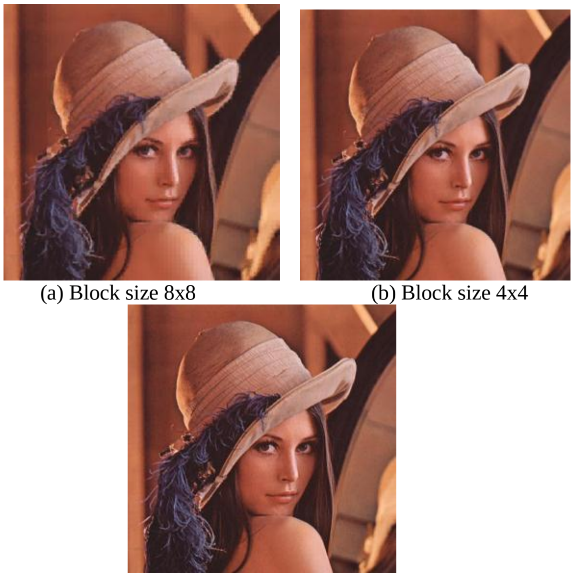
Image	Range	Bit Rate	Compression Ratio	No of Range Blocks		Encoding Time
				Smooth	Rough	
Lena	2 * 2	7.56	3.17	43927	21611	12375
	4 * 4	1.95	12.26	8394	7992	17739
	8 * 8	0.48	49.82	1559	2539	17784
Pepper	2 * 2	7.02	3.41	51409	14129	11363
	4 * 4	1.87	12.81	9851	6535	14248
	8 * 8	0.47	50.16	1638	2460	17076
Cauliflower	2 * 2	7.16	3.34	49433	16105	17989
	4 * 4	1.94	12.34	8640	7746	13016
	8 * 8	0.47	50.36	1682	2416	20484
Tajmahal	2 * 2	7.51	3.19	44672	20866	16526
	4 * 4	2.05	11.65	6624	9762	25991
	8 * 8	0.50	47.73	1018	3080	28842

Table I

a. Classification Of Blocks Compression Ratio, encoding time And Bit Rate on the Chosen Images Using the Proposed Technique

Table II

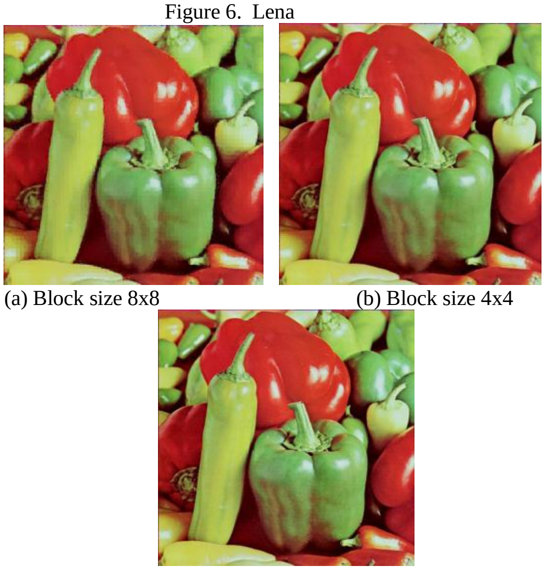
a. Classification of Blocks RMS and PSNR value On the Chosen Images Using the Proposed Technique



(a) Block size 8x8 (b) Block size 4x4

(c) Block size 2x2

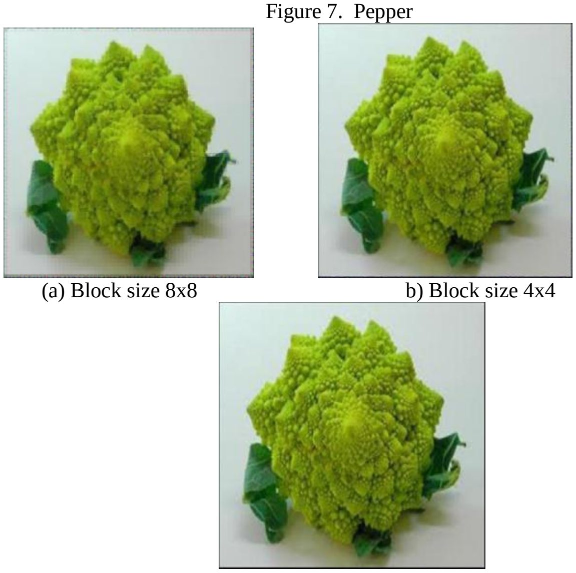
Figure 6. Lena



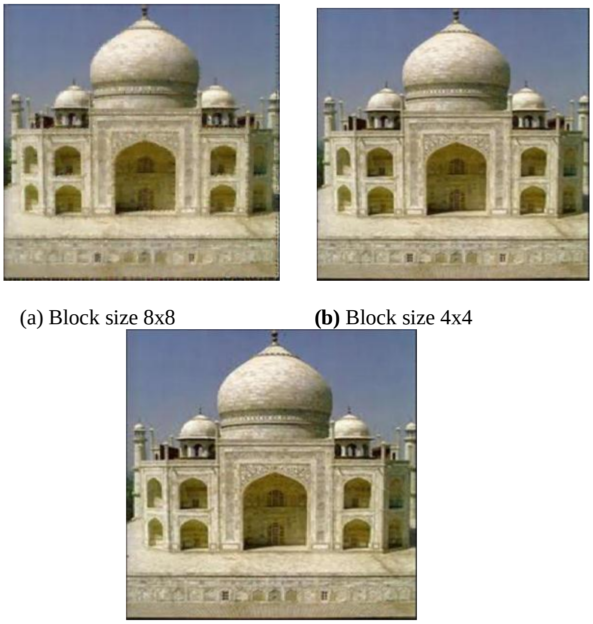
(a) Block size 8x8 (b) Block size 4x4

ISBN: 978-981-07-3161-08

(c) Block size 2x2
Figure 7. Pepper


(a) Block size 8x8 b) Block size 4x4

(c) Block size 2x2
Figure 8. Cauliflower


(a) Block size 8x8 **(b)** Block size 4x4

(c) Block size 2x2
Figure 9. Taj Mahal

Table III.

Image	Range	PSNR		
		Proposed method	Rgb & Gray Scale Component On Mpq-Btc In Image Compression[11]	Color Image Compression with Modified Fractal Coding on Spiral Architecture[12]
Lena	4 x 4	31.68	24.1209	29.05
Pepper	4 x 4	32.20	24.1531	-

a. PSNR of some methods for 512x512 images

Table III gives the result of similar methods and the proposed methods for the bench mark image of Lena and Pepper (512 x 512, 24 bit color image). In the proposed method using SA the PSNR is highly effective when compared to the existing fractal methods [11, 12].

VI. Conclusion

In this paper, a fast-encoding algorithm for fractal image coding is proposed and implemented using SA for still color image. The proposed algorithm has the better performance in terms of image quality, bit rate and coding time for RGB image. Only the encoding consumes more time but the decoding is very fast. Applications where images can be stored in a compressed form, which require faster retrieval, like medical images and photographs for identification can use the proposed method. The execution time can be further reduced by implementing the proposed method in parallel for encoding

VII. References

[1] Hsuan T. Chang and Chung J. Kuo. Iteration-free fractal image coding based on efficient domain pool design. IEEE Transaction on Image Processing, **9(3)**, pp 329-339, 2000.

[2] Y. Chakrapaniand K. Soundara Rajan, “Genetic algorithm applied to fractal image compression”, ARPN Journal of Engineering and Applied Sciences 4(1) pp 53-57, FEB 2009.

[3] Erjun Zhao Dan Liu. Fractal image compression methods: a review. Proceedings of the Third International Conference on Information Technology and Applications (ICITA’05), pp 756-759, 2005.

[4] Feng-Tse Lin, Cheng-Yan Kao, and Ching-Chi Hsu. Applying the Genetic Approach to Simulated Annealing in Solving Some NP-Hard Problems. IEEE Transactions On Systems, Man, And Cybernetics, **23(6),** 1152-1167, Nov-Dec 1993

[5] Suman K. Mitra, C. A. Murthy, and Malay K. Kundu. Technique for fractal image compression using genetic algorithm. IEEE Transaction on Image Processing, **7(4)**: 586-593, 1998.

[6] N.A .Koli and M.S. Ali,” A survey on fractal image compression key issues”, Information technology Journal 7(8): pp 1085-1095, 2008,

[7] Wohlberg B. and G. de Jager. A review of the fractal image coding literature. lEEE Transaction on Image Processing, **8**: 1716-1729, 1999.

[8] Alexandros Stamatakis. An efficient program for phylogenetic inference using simulated annealing. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS ’05) pp 1530-2075/05, 2005.

Proc. of the Intl. Conf on Advances in Computer and Information Technology - ACIT 2012

ISBN: 978-981-07-3161-08

[9] [Yung et al., 2003] Yung -Gi, Wu, Ming-Zhi, Huang, Yu-Ling, Wen. Fractal image compression with variance and mean. ICME, pp I- 353 to I-356, 2003.

[10] A.R Nadira Banu Kamal, S. Thamarai Selvi and Hennry Selvaraj“Iteration free fractal image compression using genetic algoirthm”, International Journal of Computational Intelligence and Applications, 7(4): pp 429-446, 2008.

[11] K. Somasundaram and P.Sumitra, “RGB & gray scale component on mpq-btc in image compression”, International Journal on Computer Science and Engineering,.3(4):pp 1462-1467, Apr 2011

[12] Nileshsingh V. Thakur and Dr.O.G Kakde, “Color image compression with modified fractal coding on spiral architecture” Journal of Multimedia, 2(4): pp 55-66, Aug 2007

Biography

I am a Professor of the Department of Computer Science of TBAK College for Women, Kilakarai, Ramanathaputam District, Tamil Nadu, India. I did my B Sc in Madras University, MSc in Anna University, MPhil in Alagappa University and PhD in Manonmanium Sundaranar University. My area of Interests and research are Digital Image Processing and Software Engineering. I am at present supervising 5 research scholars for their PhD degree and have guided many MPhil scholars.

.