International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 13 (2015) pp 33361-33366
© Research India Publications. http://www.ripublication.com

Fractal Image Compression Using Frog Leap Optimization Algorithm

[1] M. Sobia and [2]Dr.M.L.Valarmathi

Research Scholar, Department of CSE
Government College of Technology, Coimbatore, India
[1]sobiamaha@gct.ac.in
Associate Professor, Department of CSE
Government College of Technology, Coimbatore, India
[2]drmlv@gct.ac.in

Abstract
This paper focuses on improving fractal-based methods for image compression. Most pure fractal-based schemes are not competitive with the current state of the art, but optimization schemes incorporating fractal compression and alternative techniques have achieved considerably greater success. Frog Leap Optimization Algorithm (FLOA) is employed to optimize the mapping process. A fast and more appropriate location of the matching domain block and the corresponding transformation parameter is found for every range block. The proposed method was tested across various standard images and applied to the iteration - free method which reduces the encoding time by about 37 times compared with the conventional iteration-free method with only a slight degradation in quality.

Index Terms— Domain pool, Mean image, Frog Leap Algorithm, Optimization

I. INTRODUCTION

Representing image files as simple arrays of pixels is generally very inefficient in terms of memory. Storage space and network bandwidth is limited, so it is necessary to compress image files. There are various methods that attain very high compression ratios by letting data to be distorted in ways that are not easily perceivable to the human eye. The reduction of image data is achieved in general by the removal of redundant data. In mathematics the compression may be said as transforming the 2-D pixel array into a statistically uncorrelated data set. Usually the image compression is applied prior to the storage or transmission of the image data. Later the compressed image is decompressed to get the original image or close to original image.

Fractal image compression is a lossy compression technique. In FIC, a part of an image is designated with reference to other parts of the same image which exploits the redundancy of piecewise self-similarity. The natural images have large self-similarity. That is images contain lot of repetition. FIC finds similar patterns that exists on different scales and different places in an image, and then eliminates as much redundancy as possible. Instead of saving information for each and every similar part of the image, the system merely saves the common representation and how it is used on different scales and different places. The encoding step is very time consuming process in FIC because a large numbers of sequential search is needed to find the best match for a given range block. The iteration free FIC method uses the mean image concept where the reconstruction process is faster. Reducing the size of the domain pool and clustering the blocks to reduce the search space can reduce the encoding time. The use of FLOA for iteration-free technique results in reduction in coding time.

II. RELATED WORK

Lots of works had been conducted to speed up the process of searching the matching pairs in FIC in recent years. FIC is used to decrease the exploration cost for a range block; a portion of the domain pool is carefully chosen. For example, based on the spatial correlation on the range and domain blocks, many fast FIC schemes were proposed by searching the matching domain block from the adjacent domain block of the current range block [1] or from the adjacent domain blocks of the domain block which is the matching block for the adjacent range block of the current range block [2]. To minimize the time taken to search a range block, only part of the domain blocks are selected in variance-based accelerating [3] and region-based FIC schemes [4].

Subsequently many researchers have pursued fractal-based image compression suggesting modifications for improving the performance. Block averaging method are implemented to design the efficient domain pool based on an iteration-free fractal image codec is proposed in [5]. The redundancies between the generated domain blocks are reduced. Therefore, more efficient domain pools are obtained than those in the conventional fractal coding schemes where the coding performance is improved.

Tong and Pi [6] a new adaptive search approaches to reduce the computational complexity of fractal encoding. Bacteria Foraging Optimization Algorithm (BFOA), proposed by Passino [7], is a nature-inspired optimization algorithm. Collective intelligence algorithms are one of the most popular methods to solve these problems. All these algorithms are based on meta-heuristic methods, the most famous of which are: particle swarm optimization algorithm [8], Bee Colony Optimization Algorithm [9], Frog Leaping Algorithm [10] and Ant Colony Optimization Algorithm [11]. FLA combines the benefits of the genetic-based memetic algorithm and the social behaviour-based Particle Swarm Optimization. Recently, few papers have produced the results of the application of Shuffled Frog-Leaping Algorithm (SFLA) to various problems [12-17].

III. CONVENTIONAL FRACTAL IMAGE COMPRESSION

The image taken in conventional fractal image coding technique is divided into a number of non-overlapping range blocks. The contractive affine transformation of a gray scale maps the image of domain block D to the range block R is given by equation

$$\hat{R} = i\{\alpha(S \circ D) + \Delta g \quad (1)$$

where, $\hat{R}$ represents coded range block, i represents isometry, α represents contrast scaling, S represents contractive transformation, D represents domain block, Δg represents luminance shift.

The fractal code describing the above contractive affine transform that has minimum matching error between the original range block and coded range block are transmitted and stored. The fractal code consists of contrast scaling, luminance shift Δg, and isometry i, position P of the best matching domain block in the domain pool. In the decoding stage, an arbitrary initial image is selected and the decoded image is reconstructed by repeatedly applying the contractive affine transformation to the iterated image.

IV. AN ITERATION-FREE FRACTAL CODING

The iteration-free method proposed by Chang and Kuo [18] basically makes use of the mean image computed from the original image. In this method the M x M image is divided into blocks of size B x B. Mean values of all the range blocks are computed. A mean image is constructed from these mean values where each pixel of the mean image is the mean of the B x B range block. Hence the mean image size is M/B x M/B. Here the mean image is used as the domain pool. Due to the fact that most parts of the blocks overlap in the mean image, the neighbouring blocks have high similarity. This overlapping produces redundancies between the domain blocks. The block averaging method can reduce those redundancies. Hence, the coding performance will be improved compared with the conventional fractal schemes. The following contractive affine transformation is applied to get the matching domain block.

$$\hat{R} = i\{\alpha(D - \mu_D) + \mu_R\} \quad (2)$$

where $\hat{R}$ is the matching domain block that matches with the range block after the transformation, D is any domain block, μ_R is the mean of range block, μ_D is the mean of domain block, α is the contrast scaling, i is the isometric transformation index.

An iteration-free technique is made use of to proceed in the decoding stage. As coding of the fractal image is done with the mean image, during the decoding phase the mean image is also transmitted along with the fractal codes. To decode and obtain a particular range block of the reconstructed image, the corresponding domain block from the mean image is taken and the contractive affine transformation denoted by the fractal code is applied on it. This gives the required range block of the image in one stroke, without any iteration. Other range blocks of the image are obtained in similar manner, thereby reconstructing the original image without any iteration. Because there is no iteration, the decoding time is drastically reduced.

V. IMAGE PARTITION

There are two designs for the selection of image partition size. The square support of the input image of size M x M is partitioned into non overlapping square blocks of size B x B called range blocks. The size of the block is constant throughout the image. This scheme is called as single level partitioning of the image.

On the other hand, the square support of the input image can be partitioned into non overlapping square range cells of two different sizes, thus forming a two level square partition. The larger cells of size B x B are referred to as parent range blocks and the smaller ones normally of size B/2 x B/2 as child range blocks. The parent block can be split up into four non overlapping child blocks. The decision about the splitting of a parent block is made according to the type of the range block. This scheme is called as two level partitioning schemes. In this project single level partitioning is considered.

VI. TYPES OF BLOCK

All range blocks are classified into two classes either as a shade block or rough block according to the variance of the block. The variance of each range block can be calculated using the equation

$$Var(R) = \frac{1}{B^2} \sum_{0 \le i, j \le B} (r_{i,j} - \mu_R)^2 \quad (3)$$

where
R ⟶ Range block,
B ⟶ Size of the block
$r_{i,j}$ ⟶ Gray level value at position (i,j)
μ_R ⟶ Mean of the range block.

If the variance of the range blocks is less than a threshold value E_{th}, then the range block is of type shade block. The shade block can be approximated by a gray block, with a gray level equal to the average gray level of the block. The transformation of the shade block, which generates this uniform gray block is simply the absorption at $g_i = \mu_R$. These shades are coded simply by the mean values of the range block. All other blocks of variance greater than the threshold value are termed as rough type blocks. The rough blocks are coded by contractive affine transformation.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 13 (2015) pp 33361-33366
© Research India Publications. http://www.ripublication.com

VII. CONTRACTIVE AFFINE TRANSFORMATION

The shade blocks are coded by their mean value whereas the rough blocks are coded by the suitable domain blocks. They are searched in the domain pool and the rough type range block is mapped to a domain block with minimum distortion. Before mapping the range block, the domain block undergoes several transformations for best mapping performance. These transformations are termed as contractive affine transformation.

Since the size of the domain block is same as the size of the range block, the contraction procedure in the conventional fractal coding is eliminated. Therefore a contractive affine transformation is changed which maps the domain block to the range block. The parameters used in the new contractive transformation are specified as follows. The luminance shift in conventional schemes is to control the possible divergence in the decoding stage. The luminance shift in the conventional fractal image coding is replaced here with the mean of the range block. Hence the same mean image can be reconstructed in the decoding stage from this mean value and these mean image can be used as domain pool.

The eight isometrics are listed as,

1. Identity

2. Orthogonal reflection about mid-vertical axis (j=B-1/2) of block

$$(\tau_0\mu)_{i,j} = \mu_{i,B-1-j}$$

3. Orthogonal reflection about mid-horizontal axis (i=B-1/2) of block

$$(\tau_0\mu)_{i,j} = \mu_{B-1-i,j}$$

4. Orthogonal reflection about first diagonal (i=j) of the block

$$(\tau_0\mu)_{i,j} = \mu_{j,i}$$

5. Orthogonal reflection about second diagonal (i+j= B-1) of block

$$(\tau_0\mu)_{i,j} = \mu_{B-1-j,B-1-i}$$

6. Rotation around center of block through $+90^0$degrees

$$(\tau_0\mu)_{i,j} = \mu_{j,B-1-i}$$

7. Rotation around center of block through$+180^0$ degrees

$$(\tau_0\mu)_{i,j} = \mu_{j,B-1-i,B-1-j}$$

8. Rotation around center of block through -90^0 degrees

$$(\tau_0\mu)_{i,j} = \mu_{B-1-i,j}$$

where $\mu_{i,j}$ is the pixel value at (i, j) of the range blocks.

Then the fractal code consists of the following factors.

1. Transformation type i.e., affine transformation or mean transformation.
2. Position of the domain block which matches the range block.
3. One of the eight isometrics listed above.
4. Any of the scaling factor, α .

VIII. FROG LEAP OPTIMIZATION ALGORITHM

The approximation to the original image is obtained by extracting the fractals using the concept of fractal coding. These fractals are represented as a set of affine transformations. It has high compression ratio and simple decompression method. The main drawback is larger computational time for image compression. In order to reduce the computation time FLOA techniques have been proposed. FLOA is designed in such a way that a heuristic search is performed to achieve a global optimal solution. It is based on the development of memes carried by individuals and a global exchange of information among the population. It also unites the Particle Swarm Optimization along with its local search tool benefits and move toward a global solution by using the idea of mixing information from parallel local searches. The results are found to be very effective in attaining global solutions on testing standard images using Frog Leap (FL) algorithm.

Memeplexes are the partitioned subsets containing population of possible solutions of the FL algorithm involves defined by a set of frogs (ie. solutions). The diversity of memeplexes is considered as different cultures of frogs where a local search is performed. Within each memeplex, the process of memetic evolution gets progressed by holding ideas of individual frogs that can be influenced by the ideas of other frogs. Then ideas are passed among memeplexes in a shuffling process until convergence criteria are satisfied.

An initial population of 'P' frogs is randomly created and each frog i is represented by S variable as X_i= (X_{i1}, X_{i2}……..,X_{is}) for S-dimensional problems. According to their fitness, the frogs are sorted in a descending order. Then, the entire population is divided into m memeplexes, the second frog goes to the second memeplex, frog k goes to the k^{th} memeplex, and frog k+1 goes to the first memeplex, and so on. Within each memeplex the frogs are determined to be the best fitness as X_b and the worst fitness as X_w. X_g is recognized as global best fitness of the frog and an evolution process is applied to expand only the frog with the worst fitness (i.e. not all frogs) in every cycle. Hence, the worst fitness of frog's location is given by the equation

$$\text{Change in position}\,(D_i) = rand().(X_b - X_w) \quad (4)$$

$$New\,position\ X_w = current\ position\ X_w + D_i \quad (5)$$

where
rand () is a random number between 0 and 1

D max is the maximum allowed change in a frog's position.

From the above process the worst frog is replaced by the best frog (solution). Otherwise, the calculations in equations (4) and (5) are repeated with respect to the global best frog (i.e. X_g replaces X_b). Also with the help of arbitrary fitness, a different solution is randomly created to exchange the worst frog with the best frog. Thus a particular number of evolutionary iterations are performed and calculations are continued for each memeplex. The main parameters of the FL algorithm are: number of frogs P, number of memeplexes, and number of evolutionary iterations for each memeplex before shuffling.
The Frog Leap Optimization Algorithm (FLOP) is employed to optimize the mapping process. A fast and more appropriate location of the matching domain block and the corresponding transformation parameter is found for every range block. In this thesis, FLOA is applied to the iteration - free method which reduces the encoding time by about 37% compared with the conventional iteration-free method with only a slight degradation in quality.

IX. RESULTS AND DISCUSSION

The proposed method is tested using 256 x 256 gray scale images. To compare the performance, the following parameters are taken into account.

i. Compression time
ii. Decompression time
iii. Compression ratio
iv. PSNR (Peak Signal to Noise Ratio)

The performance of the decoded image is evaluated by the PSNR. The simulation is done in Intel core 2 Duo CPU with 2.10 GHz processing speed and with 4.00 GB RAM. Standard image like Sailboat, Mandril, Godhill, Parrot, Lena, Pepper and Dune is tested. Fig. 1. shows output images of Original, Mean and reconstructed images.
The proposed method is first tested on the Lena image with different sizes and the results are given in TABLE I. It is observed that as the population size increases, the compression time also increases, at the same time, it is seen that there is no much difference in the quality of the image for population sizes greater than 750. So the population size is taken as 750. Then, the same test images are tested with the proposed technique and the results are tabulated in Table 2. Fig. 2. shows the comparison of the original and the reconstructed images using the FLOA-based method.

TABLE I. COMPARISON RESULTS FOR DIFFERENT POPULATION SIZES FOR THE LENA IMAGE

S.No	Population Size	Compression Time (sec)	PSNR
1	50	3.40	30.38
2	100	6.43	31.29
3	150	9.59	31.88
4	200	12.79	32.20
5	250	16.02	32.55
6	300	19.13	32.70
7	350	22.36	32.93
8	400	25.52	33.09
9	450	28.74	33.17
10	500	31.86	33.25
11	550	35.12	33.43
12	600	38.28	33.51
13	650	41.83	33.60
14	700	44.69	33.60
15	750	48	33.60

TABLE II illustrates the comparison of the compression done by iteration-free method and FLOA-based method. The compression time reduction factor depicts the decrease in compression time by FLOA method, when compared to that of base iteration-free method. The maximum compression time reduction factor occurs for the Goldhil image. It has compression time reduction factor of 36.79. The minimum compression time reduction factor is 36.63 for the Dune image. The average time reduction factor is 36.71.

(a) Original Sailboat image

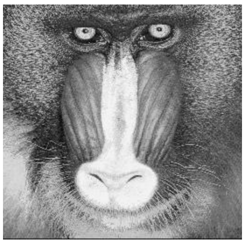
(b) Mean image

(c) Reconstructed Sailboat image

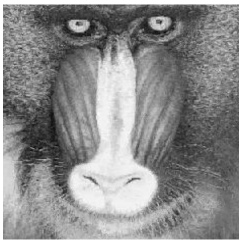
Fig. 1. Output images (a) Original Sailboat image (b) Mean image (c) Reconstructed image

TABLE II. TEST RESULTS FOR FLOA METHOD

Name of the image	Compression ratio (%)	Compression Time (sec)		Compression time reduction factor	Decompression Time (sec)		PSNR		PSNR reduction in (%)
	Iteration-free and proposed FLOA method	Iteration-free method	proposed FLOA method		Iteration-free method	proposed FLOA method	Iteration-free method	proposed FLOA method	
Lena	18.51	587.80	16.02	36.69	0.22	0.23	33.98	32.55	4.21
Dune	12.49	281.67	7.69	36.63	0.23	0.23	45.12	43.05	4.59
Parrots	16.35	477.15	12.99	36.73	0.23	0.25	32.81	31.82	3.02
Mandrill	22.06	765.61	20.87	36.68	0.23	0.23	25.65	25.12	2.07
Pepper	19.17	620.28	16.88	36.75	0.23	0.23	33.98	31.88	6.18
Sailboat	18.59	591.19	16.12	36.67	0.13	0.23	33.07	30.91	6.53
Goldhil	21.25	725.90	19.73	36.79	0.23	0.22	34.70	32.70	5.76



a) Mandrill – original image



b) Mandrill – reconstructed image

c) Goldhill - original image

d) Goldhill – reconstructed image

Fig. 2. Comparison of original and reconstructed images

The degradation in the quality of the image is measured by the difference in PSNR values of the original and the reconstructed images. The maximum reduction in quality is 6.53% which occurs for the Sailboat image. The minimum reduction in quality occurs for the Mandrill image and is 2.07%. Average reduction in quality is 4.62%. Fig. 3. Shows the comparison of compression between the conventional iteration-free method. Fig. 4. Shows the proposed method and the comparison of PSNR value.

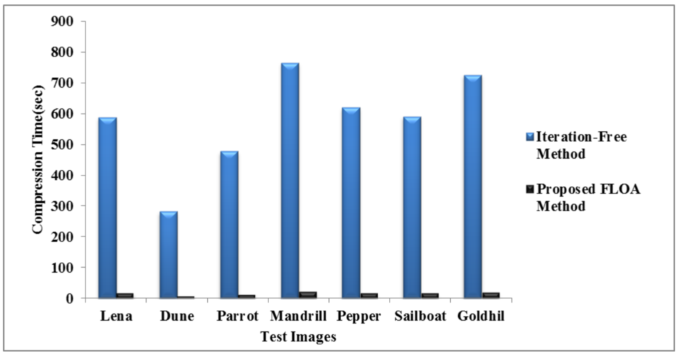


Fig. 3. Compression time for proposed and iteration-free method

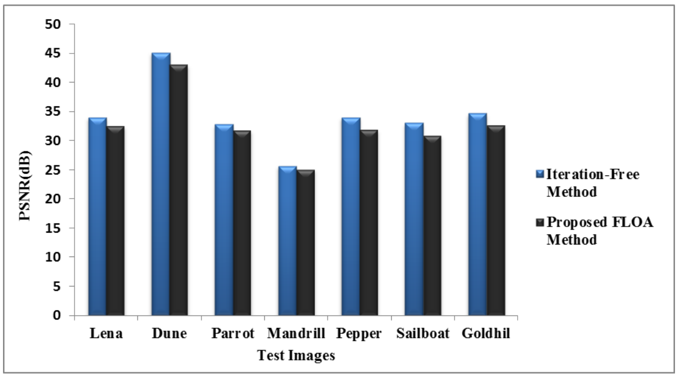


Fig. 4. PSNR value for proposed and iteration-free method

X. RESULTS AND DISCUSSION

The main objective of FLOA technique is to reduce the compression time of fractal based image compression scheme that has been tested on standard images to prove its effectiveness. Thus it ensures high quality of conventional fractal image compression technique by reducing the compression time to about thirty one times. Hence to achieve fast fractal coding and decoding of images, the use FLOA with iteration-free method is highly incorporated. Also the obtained results produce reconstructed image with good quality by reduction of PSNR value to 3.78.

REFERENCES

[1] T. K.Truong, C. M. Kung, J. H. Jeng and M. L.Hsieh, “Fast fractal image compression using spatial correlation,” Chaos, Solitons & Fractals, vol. 22, no.5, pp. 1071-1076. 2004.

[2] W. R. Schwartz and H. Pedrini “Improved fractal image compression based on robust feature descriptors,” International Journal of Image and Graphics, vol. 11, no.4, pp. 571-587, 2011.

[3] C. He, S.X. Yang and, X. Huang, “Variance-based accelerating scheme for fractal image encoding,” Electronics Letters, vol. 40, no. 2, pp. 115-116, 2004.

[4] H. Hartenstein, M. Ruhl and D. Saupe, “Region-based fractal image compression. Image processing,” IEEE Transaction on, vol. 9, no. 7, pp. 1171-1184, 2000.

[5] H. T. Chang and C. J. Kuo, “Iteration-free fractal image coding based on efficient domain pool design,” IEEE Transactions on Image Processing, vol. 9, no. 3, pp. 329-339, 2000.

[6] C.S. Tong and M. Pi, “Fast fractal image encoding based on adaptive search,” IEEE Transactions on Image processing, vol. 10, no. 9, pp. 1269-1277, 2001.

[7] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” IEEE on Control systems, vol. 22, no. 3, pp. 52-67, 2002.

[8] J. Kennedy, “Particle swarm optimization,” In Encyclopedia of Machine Learning Springer US, pp. 760-766, 2010.

[9] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," Applied Mathematics and Computation, vol. 214, no. 1, 108-13, 2009.

[10] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," Journal of Water Resources Planning and Management, vol. 129, no. 3, 210-225, 2003.

[11] M. Dorigo, and M. Birattari, "Ant colony optimization. In Encyclopaedia of machine learning," Springer US, pp. 36-39, 2010.

[12] X. H. LUO, Y.Yang and X. Li, "Modified shuffled frog-leaping algorithm to solve traveling salesman problem," Journal on Communications, vol. 30, no. 7, pp. 130-135, 2009.

[13] L.P. Xue, J.X, Yin, J.R. Zhou and Z. Ji, "A novel shuffled particle-pair optimizer for speaker recognition," Journal of Electronics & Information Technology, vol. 6, pp. 1359-1362, 2009.

[14] M. Eusuff, K. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Engineering Optimizatio," vol. 38, no. 2, pp. 129-154, 2006.

[15] A. Rahimi-Vahed, M. Dangchi, H. Rafiei and E. Salimi, "A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem," The International Journal of Advanced Manufacturing Technology, vol. 41, no. 12, pp. 1227-1239, 2009.

[16] A. Khorsandi, A. Alimardani, B. Vahidi and S. H. Hosseinian, "Hybrid shuffled frog leaping algorithm and Nelder–Mead simplex search for optimal and Nelder–Mead simplex search for optimal reactive power dispatch," IET generation, transmission & distribution, vol. 5, no. 2, pp. 249-256, 2011.

[17] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," Advanced Engineering Informatics, vol. 19, no. 1, pp. 43-53, 2005.

[18] H. T.Chang and C. J. Kuo, "Iteration-free fractal image coding based on efficient domain pool design," IEEE Transactions on Image Processing, vol. 9, no. 3, pp. 329-339, 2000.