Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 2 (2010) 338–344

Procedia Computer Science

www.elsevier.com/locate/procedia

ICEBT 2010

Adaptive Fractal Image Compression using PSO

A.Muruganandham[a, *], Dr. R.S.D. Wahida Banu[b]

[a]*Sona College of Technology, Salem, India*
[b]*Govt Engineering College, Salem, India*

Abstract

At low bitrate and with acceptable quality in Fractal Image Compression (FIC) of the coded image, the encoding time is very large for most existing algorithms. In this paper, a fast fractal encoding system is proposed using particle swarm optimization (PSO) to reduce the encoding time. Here, an optimization technique is used for the MSE based on the stopping criterion between range block and domain block. This PSO technique can speedup the fractal encoder and preserve the image quality for medical imaging.

Keywords- Fractal Image Compression (FIC);Mean Square Error (MSE) ;Particle Swarm Optimization (PSO); Iteration Function System (IFS).

1. Introduction

The idea of the image redundancies can be efficiently exploited by means of, block self-affine transformations may call the fractal image compression (FIC). The first practical fractal image compression scheme was introduced in 1992 by Jacquin [3, 4]. The fractal transform for image compression was introduced in 1985 by Barnsley and Demko [1, 2]. The very high encoding time is the main disadvantages because of exhaustive search strategy. Therefore, decreasing the encoding time is an interesting research topic for FIC [3, 4].

One way of decreasing the encoding time is by using stochastic optimization methods using Genetic Algorithm (GA) this recent topics of GA-based methods are proposed to improve the efficiency [5, 6]. The idea of special correlation of an image is used in these methods, while the chromosomes in GA consist of all range blocks which leads to high encoding speed.

Other researchers focused on improvements by tree structure search methods [12, 13]of the search process and parallel search methods [14, 15] or quad tree partitioning of range blocks [9,16] to make it faster.

In this paper present a fast fractal encoding using particle swarm optimization. The outline of the remaining part of this paper is as follows: Section II includes fractal image coding. Section III involves implementation of PSO search. Section IV concerns the proposed fast fractal encoder using PSO optimization, and in Section V experimental results and comparisons are included. In Section VI present our conclusions.

2. Fractal Image Compression Algorithm

The fundamental idea of fractal image compression is based on an Iteration Function System (IFS) in which the governing theorems are the Collage Theorem and the Contractive Mapping Fixed-Point Theorem [7]. The encoding unit of FIC for given gray level image of size N x N is $(N/L)^2$ of non-overlapping range blocks of size L x L which forms the range pool *R*. For each range block *v* in *R,* one search in the $(N - 2L + 1)^2$ overlapping domain blocks of size 2L x 2L which forms the domain pool *D* to find the best match. The parameters describing this fractal affine transformation of domain block into range block form the fractal compression code of *v*.

* Corresponding Author. Tel.:+91-9443825711
E-mail address: muruga_salem@rediffmail.com

1877-0509 © 2010 Published by Elsevier Ltd Open access under CC BY-NC-ND license.
doi:10.1016/j.procs.2010.11.044

The parameters of fractal affine transformation is Φ of domain block into range block having domain block coordinates (t_x, t_y), Dihedral transformation-*d,* contrast scaling-*p,* brightness offset-*q.* The affine transformation is illustrated as flowchart in fig. 1 for this fractal.

$$\phi\begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & p \end{bmatrix}\begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ q \end{bmatrix}, \tag{1}$$

Where the 2 x 2 sub-matrix $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is one of the Dihedral transformations in (2)

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_3 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad T_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad T_5 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$
$$T_6 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad T_7 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{2}$$

The above parameters are found using the following procedure

1. the domain block is first down-sampled to L x L and denoted by *u*
2. The down-sampled block is transformed subject to the eight transformations T_k: k = 0,. . . ,7 in the Dihedral on the pixel positions and are denoted by u_k, $k = 0,1, \ldots ,7$, where $u_0 = u$. The transformations T_1 and T_2 correspond to the flips of *u* along the horizontal and vertical lines, respectively. T_3 is the flip along both the horizontal and vertical lines. T_4, T_5, T_6, and T_7 are the transformations of T_0, T_1, T_2, and T_3 performed by an additional flip along the main diagonal line, respectively.
3. For each domain block, there are eight separate MSE computations required to find the index d such that

$$d = \arg\min\{\text{MSE}((p_k u_k + q_k), v) : k = 0,1,\ldots,7\} \tag{3}$$

Where $$\text{MSE}(u,v) = \frac{1}{L^2}\sum_{i,j=0}^{L-1}(u(i,j=0) - v(i,j))^2 \tag{4}$$

Here, p_k and q_k can be computed directly as

$$p_k = \frac{\left[L^2\langle u_k, v\rangle - \sum_{i=0}^{L-1}\sum_{j=0}^{L-1}u_k(i,j)\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}v(i,j)\right]}{\left[L^2\langle u_k, u_k\rangle - \left(\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}u_k(i,j)\right)^2\right]}, \tag{5}$$

$$q_k = \frac{1}{L^2}\left[\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}v(i,j) - p_k\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}u_k(i,j)\right] \tag{6}$$

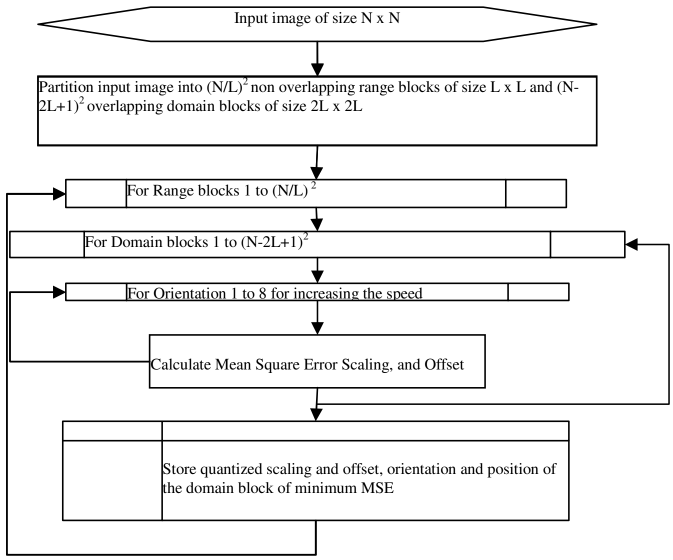

Fig. 1 Fractal affine transformation of domain block into range block

4. As u runs over all of the domain blocks in D to find the best match, the terms t_x and t_y can be obtained together with d and the specific p and q corresponding this d, the affine transformation (1) is found for the given range block v.

In practice, t_x, t_y, d, p, and q can be encoded using $\log_2$ (N), $\log_2$ (N), 3, 5, and 7 bits, respectively, which are regarded as the compression code of v. Finally, the encoding process is completed as v runs over all of the (N/L)^2 range blocks in R.

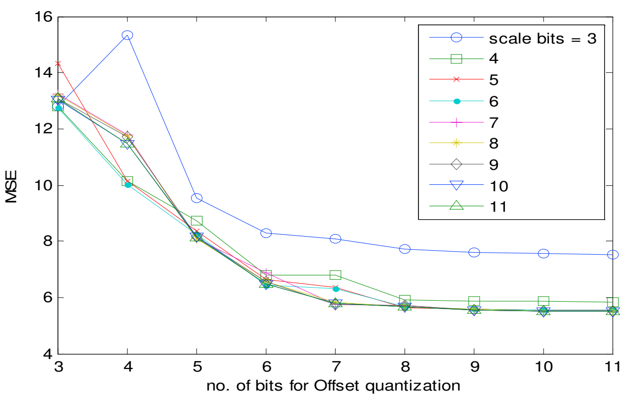

Fig. 2 MSE Vs. quantization parameter

From fig. 2 find the MSE vs. quantization parameter for randomly selected range block of size 8 x 8 from 256 x 256 Lena image. From fig. 2, choosing 5 bits and 7 bits as quantization parameter for scale and offset value are justified respectively.

To decode, the compression codes to obtain a new image, and proceeds recursively by chooses any image as the initial one and makes up the (N/L)^2 affine transformations. According to Partitioned Iteration Function Theorem (PIFS), the sequence of images will converge. According to user's application the stopping criterion of the recursion is designed based on the PSO. The final image is the retrieved image of fractal coding.

3. Particle Swarm Optimization (PSO)

A population-based algorithm is PSO for searching global optimum. To simulate a simplified social behaviour [8, 9] is the way of original idea of PSO. Similar to the crossover operation of the GA, in PSO the particles are adjusted toward the best individual experience (PBEST) and the best social or global experience (GBEST). However, PSO is unlike a GA, why because in that each potential solution, particle is ‘‘flying” through hyperspace with a velocity, the particles and the swarm have memory for process; in the population of the GA memory does not exist.

Let $x_{j,d}(t)$ and $v_{j,d}(t)$ denote the d^{th} dimensional value of the vector of position and velocity of j^{th} particle in the swarm, respectively, at time t. The PSO model can be expressed as

$$v_{j,d}(t) = v_{j,d}(t-1) + c_1.\varphi_1.(x^*_{j,d} - x_{j,d}(t-1)) + c_2.\varphi_2.(x^\#_d - x_{j,d}(t-1)), \quad (7)$$

$$x_{j,d}(t) = x_{j,d}(t-1) + v_{j,d}(t), \quad (8)$$

where $x^*_{j,d}$ (PBEST) denotes the best position of j^{th} particle up to time t-1 and $x^\#_d$ (GBEST) denotes the best position of the whole swarm up to time t-1, φ_1 and φ_2 are random numbers, and c_1 and c_2 represent the individuality and sociality coefficients, respectively.

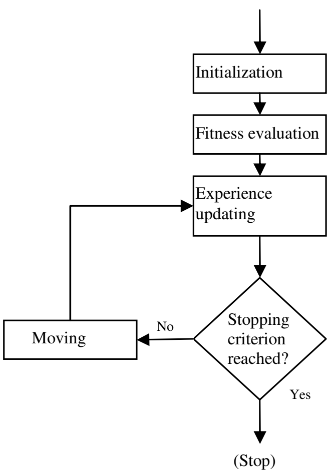

Fig. 3 Block diagram of PSO

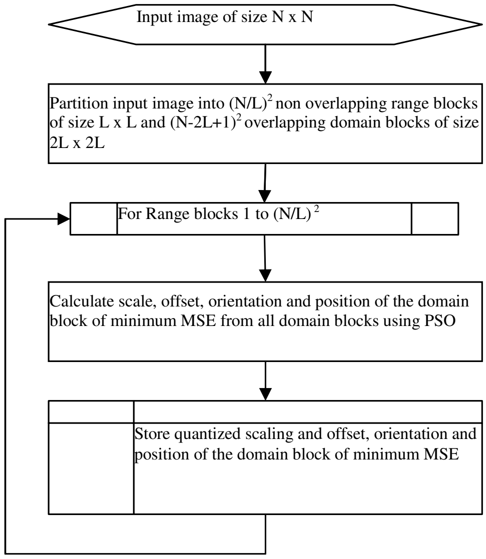

Fig.4 Fractal encoding of proposed method

The steps involved here is the population size is first determined, and the velocity and position of each particle are initialized. Each particle moves according to (7) and (8), and the fitness is then calculated. Meanwhile, the best positions of each swarm and particles are recorded. Finally, as the stopping criterion is satisfied, the best position of the swarm is the final solution. The block diagram of PSO is displayed in Fig. 3 and the main steps are given as follows:

1. Set the swarm size. Initialize the velocity and the position of each particle randomly.
2. For each j, evaluate the fitness value of x_j and update the individual best position $x^*_{j,d}$ if better fitness is found.
3. Find the new best position of the whole swarm. Update the swarm best position $x^\#$ if the fitness of the new best position is better than that of the previous swarm.
4. If the stopping criterion is satisfied, then stop.
5. For each particle, update the position and the velocity according to (8) and (7). Go to step 2.

4. Proposed Method

In the proposed fast fractal encoding using PSO, reduce the encoding time by reducing the searching time to find a best match domain block for the given range block from all domain blocks.

Flowchart of the fractal encoding of the proposed method is shown in fig. 4.

Domain block of minimum MSE and stopping criterion is found by PSO using the steps given below:

1. Set the swarm size must be proportional to $(N\text{-}2L\text{+}1)^2$/(maximum no. of iterations for PSO) and initialize the each particle velocity and the position randomly

2. By using eqn. (3) the fitness value includes finding , MSE between domain block specified by the particles position and given range block
3. Update swarm best position if the fitness of the new best position is better than that of the previous swarm.
4. If swarm best position is not changed for some percentage of maximum iteration for PSO, then stop
5. The best position of the particles is updated using eqn. (7) and (8) and go to step 2.

5. Experimental Results

The results have been compared to the full search FIC mentioned in the previous sections in terms of encoding time and PSNR of fast fractal encoding using PSO

Figures of 5a, 5b, 5c shows the original images Lena, Goldhill, Cameraman 256 x 256 at 0.5bpp and Figures of 6a, 6b, 6c shows the decoded Lena, Goldhill, Cameraman image using full search FIC and fast fractal encoding using PSO.

Fig. 5a.Original 256 x 256 Lena Image

(a) (b)

Fig. 6. Decoded Lena image at 0.5 bits per pixel for (a) Full search FIC (b) proposed fast fractal encoding using PSO

Figure 9c shows the variation in PSNR by varying the stopping criterion in fast fractal encoding using PSO by changing the percentage of maximum iteration of PSO. From figure 9c the variation of PSNR with variation of stopping criterion is very less. Hence a 10% of maximum iteration for PSO is chooses as stopping criterion with example as shown from the figure (9a, 9b, 9d).for various swarm size.

Fig. 5b. Original 256 x 256 Goldhill Image

(a) (b)

Fig. 7. Decoded Goldhill image at 0.5 bits per pixel for (a) Full search FIC b) proposed fast fractal encoding using PSO

Fig. 5c. Original 256 x 256 Camera man Image

(a) (b)

Fig. 8. Decoded Camera man image at 0.5 bits per pixel for (a) Full search FIC (b) proposed fast fractal encoding using PSO

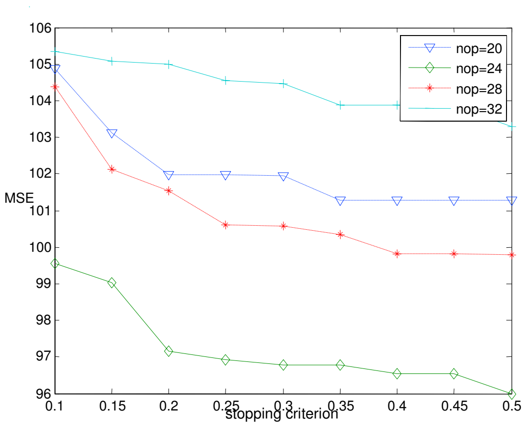

Figure 9a.
MSE vs. Stopping criterion for Lena image512 x 512

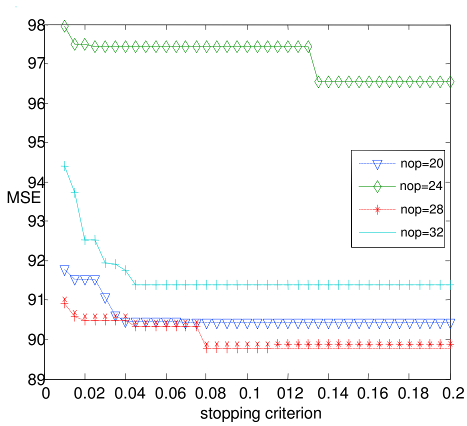

Figure 9b.
MSE vs. Stopping criterion for camera man512 x 512

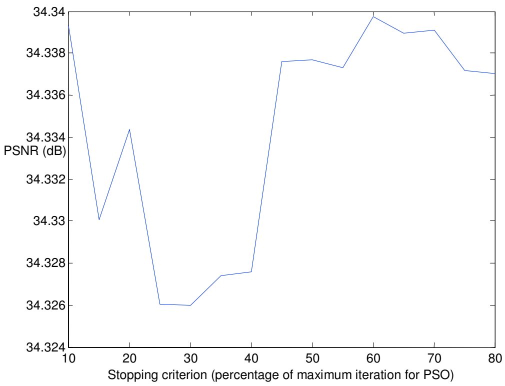

Figure 9c.
variation in PSNR by varying the stopping criterion

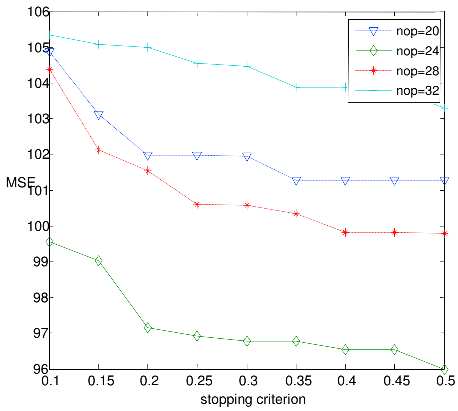

Figure 9d.
MSE vs. Stopping criterion for Goldhill512 x 512

The numeric results containing bitrates, encoding time and PSNR of decoded image of various images are tabulated in table I.

TABLE I comparison results of encoding time and PSNR of Lena, Goldhill, Camera man images

Input image	Bitrates (bpp)	Method	Encoding Time (hh: mm: ss)	PSNR (dB)
Lena	0.5	Full search FIC	09:07:20	35.80
		Proposed method	00:15:34	35.03
Goldhill	0.5	Full search FIC	09:02:12	33.64
		Proposed method	00:17:37	32.77
Camera man	0.5	Full search FIC	09:02:49	35.11
		Proposed method	00:15:24	34.23

6. Conclusion

Fractal image compression can produce better compression ratio at acceptable quality. By using PSO for fractal coding can reduce the encoding time with 1.2db loss in image quality. This can be improved by applying some classification technique and various FIC methods. Due to the fast encoding reasons this may be used in medical imaging (Through Internet Data Transfer)

References

[1] M.F. Barnsley, S. Demko, Iterated function systems and the global construction of fractals, Proc. Roy. Soc. Lond A399 (1985) 243–275.

[2] A.E. Jacquin, Fractal image coding: a review, Proc. IEEE 10 (1993)1451–1465.

[3] A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Transactions on Signal Processing 1 (1992) 18–30.

[4] M.F. Barnsley, A.D. Sloan, A better way to compress images, BYTE Magazine (1988) 215–233

[5] M. Polvere, M. Nappi, Speed-up in fractal image coding: comparison of methods, IEEE Transactions on Image Processing 9 (2000) 1002–1009.

[6] T.K. Truong, J.H. Jeng, I.S. Reed, P.C. Lee, A.Q. Li, A fast encoding algorithm for fractal image compression using the DCT inner product, IEEE Transactions on Image Processing 9 (4) (2000) 529–535.

[7] M.S. Wu, J.H. Jeng, J.G. Hsieh, Schema genetic algorithm for fractal image compression, Engineering Applications of Artificial Intelligence 20 (2007) 531–538.

[8] M.S. Wu, W.C. Teng, J.H. Jeng, J.G. Hsieh, Spatial correlation genetic algorithm for fractal image compression, Chaos Solitons & Fractals 28 (2) (2006) 497–510.

[9] Y. Fisher, *Fractal Image Compression—Theory and Application*. New York: Springer-Verlag, 1994.

[10] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, 1995, pp. 1942–1948.

[11] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of IEEE International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.

[12] B. Bani-Eqbal, Speeding up fractal image compression, Proc. SPIE: Still-Image Compress. 2418 (1995) 67–74.

[13] B. Hurtgen, C. Stiller, Fast hierarchical codebook search for fractal coding still images, SPIE Visual Commun. PACS Med. Appl. (1993) 397–408.

[14] C. Hufnagl, A. Uhl, Algorithms for fractal image compression on massively parallel SIMD arrays, Real-Time Imaging 6 (2000) 267–281.

[15] D. Vidya, R. Parthasarathy, T.C. Bina, N.G. Swaroopa, Architecture for fractal image compression, J. Syst. Archit. 46 (2000) 1275–1291.

[16] Y. Fisher, Fractal image compression, SIGGRAPH'92 Course Notes 12 (1992) 7.1–7.19.