ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 6 (2010) No. 1, pp. 40-46

Implementation of fractal image compression employing particle swarm optimization

Y. Chakrapani[1]* , K. Soundararajan[2]

[1] Department of ECE, G. Pulla Reddy Engineering College, Kurnool 518002, India
[2] Department of ECE, J. N. T. U College of Engineering, Anantapur 515002, India

(*Received June 26 2009, Accepted November 12 2009*)

Abstract. In this paper the technique of particle swarm optimization (PSO) is applied for fractal image compression (FIC). With the help of this evolutionary algorithm effort is made to reduce the search complexity of matching between range block and domain block. One of the image compression techniques in the spatial domain is Fractal Image Compression but the main drawback of FIC is that it involves more computational time due to global search. In order to improve the computational time and also the acceptable quality of the decoded image, PSO algorithm is proposed. Experimental results show that the PSO is a better method than the traditional exhaustive search method in terms of encoding time.

Keywords: fractal image compression, particle swarm optimization, Pbest, Gbest, encoding time

1 Introduction

Compression and decompression technology of digital image has become an important aspect in the storing and transferring of digital image in information society. Most of the methods in use can be classified under the head of lossy compression. This implies that the reconstructed image is always an approximation of the original image. Fractal image coding introduced by Barnsley and Jacquin[1, 4, 5] is the outcome of the study of the iterated function system developed in the last decade. Because of its high compression ratio and simple decompression method, many researchers have done a lot of research on it. But the main drawback of their work can be related to large computational time for image compression. At present, researchers focus mainly on how to select and optimize the classification of the range blocks, balance the speed of compression, increase the compression ratio and improve the quality of image after decompression[9]. Especially in the field of reducing the complexity of search, many outstanding algorithms based on classified search have been proposed[3]. One attractive feature of fractal image compression is that it is resolution independent in the sense that when decompressing, it is not necessary that the dimensions of the decompressed image be the same as that of original image.

Particle swarm optimization (PSO) is suggested by Eberhart and Kennedy based on the analogy of swarm of birds and school of fish[6]. PSO mimics the behavior of individuals in a swarm to maximize the survival of the species. In PSO, each individual makes his decision using his own experience together with other individuals' experiences. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of moving points in a multidimensional space. The individual particles are drawn stochastically toward the position of present velocity of each individual, their own previous best performance, and the best previous performance of their neighbors[7, 8]. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational

* Corresponding author. *E-mail address*: yerramsettychakri@gmail.com.

efficiency when compared with mathematical algorithms and other heuristic optimization techniques[10]. PSO can be easily applied to nonlinear and non-continuous optimization problems.

Even though a few investigations have been carried out in application of evolutionary techniques to fractal image compression[2], no work has been reported on the application of PSO based FIC. The remainder of the paper is organized in detail as follows: Section 2 focuses on the theory of transformations and fractal image compression technique. In Section 3 the concept of PSO and its application to FIC is explained. Section 4 and Section 5 deals with experimental results and discussions. In Section 6 some conclusions are drawn.

2 Fractal image compression

The fractal image compression algorithm is based on the fractal theory of self-similar and self-affine transformations.

2.1 Self-affine and self-similar transformations

In this section we present the basic theory involved in Fractal Image Compression. It is basically based on fractal theory of self-affine transformations and self-similar transformations. A self-affine transformation $W : R^n \rightarrow R^n$ is a transformation of the form $W(x) = T(x) + \boldsymbol{b}$, where T is a linear transformation on R^n and $\boldsymbol{b} \in R^n$ is a vector.

A mapping $W : D \rightarrow D, D \subseteq R^n$ is called a contraction on D if there is a real number c, $0 < c < 1$ such that $d(W(x), W(y)) \leq cd(x, y)$ for $x, y \in D$ and for a metric d on R^n. The real number c is called the contractivity of W.

$$d(W(x), W(y)) = cd(x, y),$$

then W is called a similarity.

A family $\{w_1, \cdots, w_m\}$ of contractions is known as Local Iterated function scheme (LIFS). If there is a subset $F \subseteq D$ such that for a LIFS $\{w_1, \cdots, w_m\}$.

$$F = \cup_{i=1}^{m} w_i(F). \tag{1}$$

Then F is said to be invariant for that LIFS. If F is invariant under a collection of similarities, F is known as a self-similar set. Let denote the class of all non-empty compact subsets of D. The $\delta-$ parallel body of $A \in S$ is the set of points within distance δ of A, i.e

$$A_\delta = \{x \in D : |x - a| \leq \delta, a \in A\}. \tag{2}$$

Let us define the distance $d(A, B)$ between two sets A, B to be

$$d(A, B) = \inf\{\delta : A \subset B_\delta \wedge B \subset A_\delta\}.$$

The distance function is known as the Hausdorff metric on S. We can also use other distance measures. Given a LIFS $\{w_1, \cdots, w_m\}$, there exists an unique compact invariant set F, such that $F = \bigcup_{i=1}^{m} w_i(F)$, this F is known as attractor of the system. If E is compact non-empty subset such that $w_i(E) \subset E$ and

$$W(E) = \cup_{i=1}^{m} w_i(F). \tag{3}$$

We define the k-th iteration of W, $W^k(E)$ to be $W^0(E) = E, W^k(E) = W(W^{(k-1)}(E))$ for $K \geq 1$ then we have

$$F = \bigcap_{i=1}^{\infty} W^k(E). \tag{4}$$

The sequence of iteration $W^k(E)$ converges to the attractor of the system for any set E. This means that we can have a family of contractions that approximate complex images and, using the family of contractions, the images can be stored and transmitted in a very efficient way. Once we have a LIFS it is easy to obtain the encoded image.

If we want to encode an arbitrary image in this way, we will have to find a family of contractions so that its attractor is an approximation to the given image. Barnsley's Collage Theorem states how well the attractor of a LIFS can approximate the given image.

Theorem 1. *Let* $\{w_1, \cdots, w_m\}$ *be contractions on* R^n *so that* $|w_i(x) - w_i(y)| \le c|x-y|, \forall x, y \in R^n \wedge \forall i$, *where* $c < 1$. *Let* $E \subset R^n$ *be any non-empty compact set. Then*

$$d(E, F) \le d\left(E, \bigcup_{i=1}^{m} w_i(E)\right) \frac{1}{(1-c)}. \tag{5}$$

Where F *is the invariant set for the* w_i *and* d *is the Hausdorff metric.*

As a consequence of this theorem, any subset R^n can be approximated within an arbitrary tolerance by a self-similar set; i.e., given $\delta > 0$ there exist contacting similarities $\{w_1, \cdots, w_m\}$ with invariant set F satisfying $d(E, F) < \delta$. Therefore the problem of finding a LIFS $\{w_1, \cdots, w_m\}$ whose attractor F is arbitrary close to a given image I is equivalent to minimizing the distance $d\left(I, \bigcup_{i=1}^{m} w_i(I)\right)$.

2.2 Fractal image coding

The main theory of fractal image coding is based on iterated function system, attractor theorem and Collage theorem. Fractal Image coding makes good use of Image self-similarity in space by ablating image geometric redundant. Fractal coding process is quite complicated but decoding process is very simple, which makes use of potentials in high compression ratio. Fractal Image coding attempts to find a set of contractive transformations that map (possibly overlapping) domain cells onto a set of range cells that tile the image. One attractive feature of fractal image compression is that it is resolution independent in the sense that when decompressing, it is not necessary that the dimensions of the decompressed image be the same as that of original image.

The basic algorithm for fractal encoding is as follows:

- The image is partitioned into non overlapping range cells $\{R_i\}$ which may be rectangular or any other shape such as triangles. In this paper rectangular range cells are used.
- The image is covered with a sequence of possibly overlapping domain cells. The domain cells occur in variety of sizes and they may be in large number.
- For each range cell the domain cell and corresponding transformation that best covers the range cell is identified. The transformations are generally the affined transformations. For the best match the transformation parameters such as contrast and brightness are adjusted as shown in Fig. 1.
- The code for fractal encoded image is a list consisting of information for each range cell which includes the location of range cell, the domain that maps onto that range cell and parameters that describe the transformation mapping the domain onto the range.

3 Particle swarm optimization

Particle swarm optimization is a computation technique developed by Eberhart and Kennedy based on the analogy of swarm of birds and school of fish. PSO mimics the behavior of individuals in a swarm to maximize the survival of the species. It is similar to other evolutionary computation techniques like Genetic Algorithm (GA) in conducting searching for optima using an initial population of individuals. The individuals of this initial population are then updated according to some kind of process such that they are moved to a

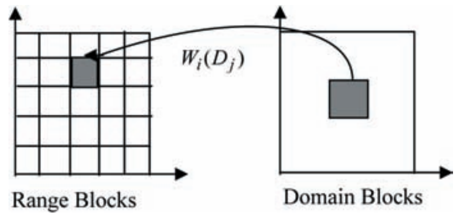

Fig. 1. Domain-range block transformations

better solution area. GA is motivated by evolution seen in nature and it borrows the principle of competition and survival of fittest from the concept of evolution. PSO on the other hand is motivated from the simulation of social behavior. It borrows the principle of cooperation and competition among the individuals themselves. However this approach is advantageous over GA in more than one way. First, PSO has memory. That is, every particle remembers its best solution (Pbest) as well as the group best solution (Gbest). Another advantage of PSO is that the initial population of PSO is maintained and so there is no need for applying operators to the population, a process which is time and memory-storage consuming.

3.1 Overview of PSO

In PSO system, each individual adjusts its flying in a multi-dimensional search space according to its own flying experience and its companions flying experience. Each individual is referred to as "particle" which represents a candidate solution to the problem. Each particle is treated as a point in a D-dimensional space. The i^{th} particle is represented as $X_j = (x_1, x_2, x_3, \cdots, x_j)$. The best position (giving the minimum fitness value) of each particle is recorded and represented as $Pbest_j = (P_1, P_2, P_3, \cdots, P_j)$. The position of the best particle among all the values in the swarm for that iteration is considered as Gbest. The velocity for particle 'i' is represented as $V_j = (V_1, V_2, V_3, \cdots, V_j)$. Using the information, the velocity and position of each particle is updated according to Eq. (4), Eq. (5) and Eq. (4), Eq. (6).

$$V_i(n+1) = V_i(n) = c_1 r_1(\text{Pbest}_i^n - X_i) + c_2 r_2((Gbest) - X_i). \quad (6)$$

$$X_i(n+1) = X_i(n) + V_i(n+1). \quad (7)$$

Where c_1 and c_2 are acceleration coefficients and r_1 and r_2 are uniformly distributed random numbers in the range (0,1).

3.2 Implementation of PSO based FIC

In a typical run of the PSO, for every range block, an initial swarm of random values which correspond to the top left coordinates of domain blocks and its isometry are generated. Each random value corresponds to the location of the domain block and is used to evaluate the domain block and find the MSE. The domain block with the minimum MSE in the swarm is identified and its coordinates are noted as Pbest values. Each particle keeps track of its coordinates in hyperspace which are associated with the fittest solution it has achieved so far. The Pbestdomain of each particle at iteration is updated according to Eq. (8).

$$\text{Pbestdomain}_i^n = \begin{cases} X_i^n & \text{if } f_i(n) \leq f_i(n-1) \\ \text{Pbestdomain}_i^{n-1} & \text{otherwise} \end{cases} \quad (8)$$

Where $f_i(n)$ is the Mean Squared error (MSE) value between the range block and selected domain block in the present iteration. The global best value Gbestdomain is set as the best evaluated position among all Pbestdomain_i^n in that iteration . The velocity and position of each particle are updated using Eq. (6) and Eq. (7) in each iteration. The application of PSO involves repeatedly performing two steps:

- The calculation of the objective function (MSE) for each of the particle in the current population 'i'.
- The particle swarm optimization then updates the particle coordinates based on Eq. (6) and Eq. (7)

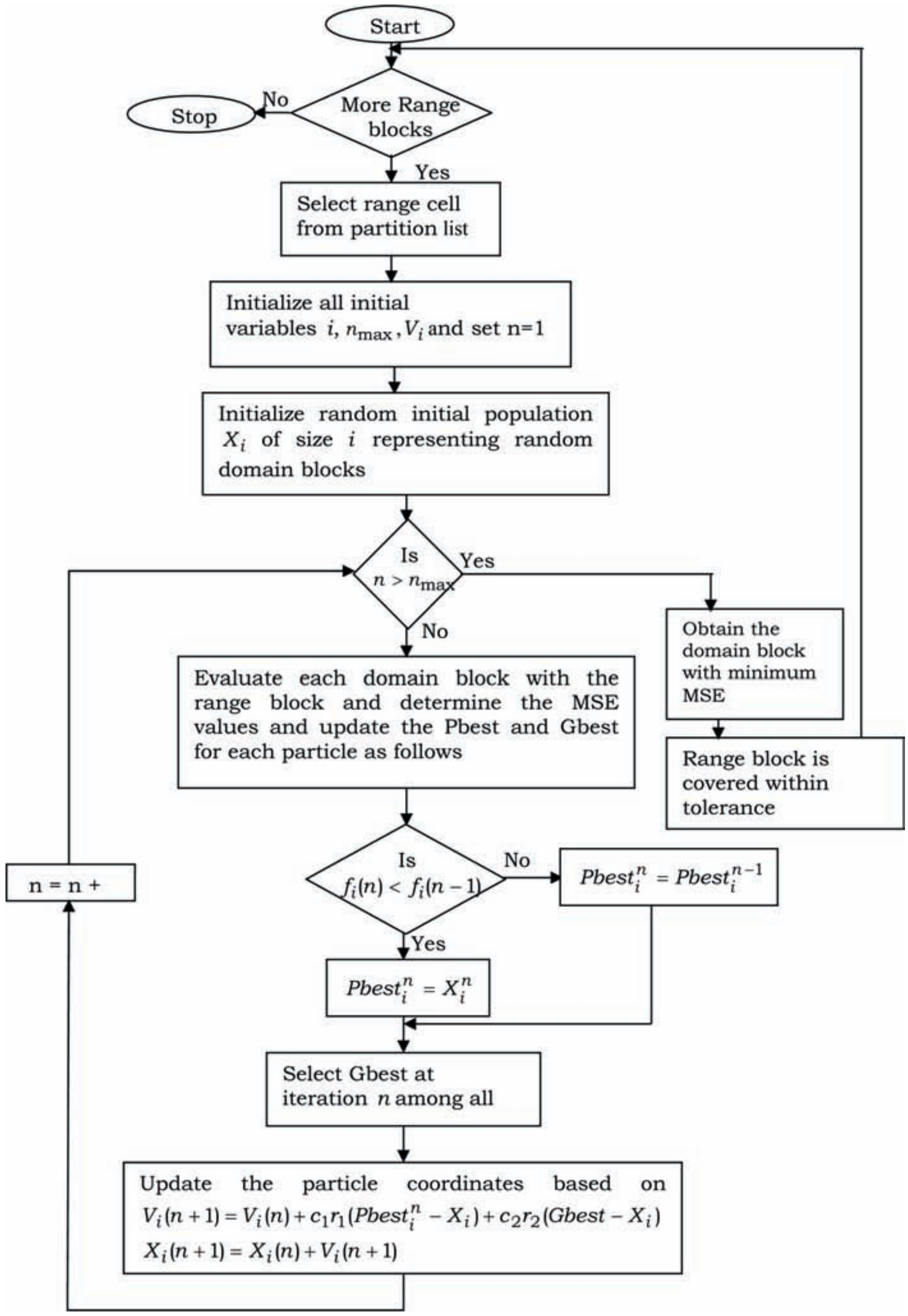

Fig. 2. Domain-range block transformations

These two steps are repeated from population to population until a stopping criterion terminates the search. At the end of last iteration the Gbestdomain value is noted from the Pbestdomain values and utilizing the Gbestdomain value, the domain block matching for the range block is done. Fig. 2 shows the flowchart representation of PSO based FIC. The various parameters of PSO are given in Tab. 1:

4 System investigated

In this paper a Gray level image of 256×256 size with 256 Gray levels is considered. A Range block of size 4×4 and Domain blocks of size 8×8 are considered. The domain blocks are mapped to the range block by affine transformations and the best domain block is selected. The mean squared error (MSE) and PSNR considered in this work are given by:

Table 1. Parameters of particle swarm optimization

Range block Size	4×4	Swarm size	50
Acceleration coefficient	4	Iterations f	25

Table 2. Comparison of exhaustive search based FIC and PSO based FIC

Image	PSNR (dB)		Compression ratio (bpp)		Encoding time (sec)	
	FIC	PSO based FIC	FIC	PSO based FIC	FIC	PSO based FIC
Lena	35.26	34.39	0.98	1.89	8600	6500
Barbara	34.674	32.98	0.98	1.89	8400	6620

$$\text{MSE} = \frac{1}{N_{\text{Rows}} N_{\text{Cols}}} \sum_{i=1}^{N_{\text{Rows}}} \sum_{j-1}^{N_{\text{Cols}}} |f_{i,j} - d_{i,j}|^2 \quad \text{and} \quad \text{PSNR} = 10 \log_{10} \left[\frac{255^2}{\text{MSE}} \right]. \tag{9}$$

5 Results and discussions

This work is carried out in MATLAB 7.0 version on Pentium-4 processor with 1.73 GHz and 256 MB RAM and the original image is classical 128×128 Lena and Barbara face image coded with 8 bits per pixel. A random swarm of points are generated and each point is evaluated in the following manner. The point is converted into its corresponding binary value and the first 16 bits are utilized to locate the top left corner of domain block and next 3 bits are used to find the isometry to be applied to the selected domain block. The points in the swarm are evaluated as explained in Section 3.2 until the convergence criteria is satisfied. Tab. 2 shows the comparison of PSO based FIC with the traditional exhaustive search method. It can be seen from the table that the PSNR and encoding time with the proposed technique has been improved as compared with the traditional method. Fig. 3 and Fig. 4 show the reconstructed images using FIC with PSO as search algorithm along with the original images of Barbara and Lena after 25 iterations.

Fig. 3. Original image of Lena and Reconstructed image of Lena

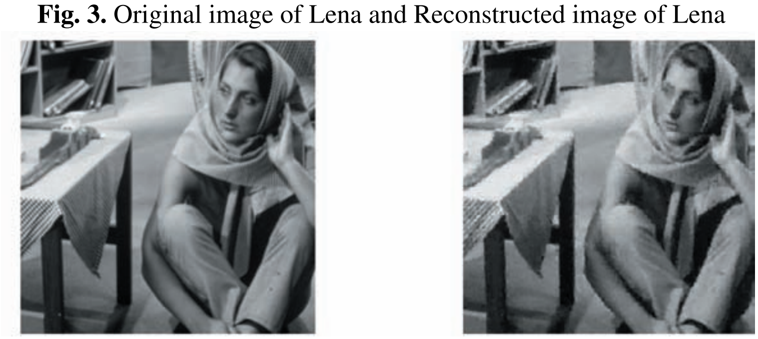

Fig. 4. Original image of Barbara and Reconstructed image of Barbara

6 Conclusions

In this paper the concept of PSO is applied to FIC. Instead of global searching in FIC the evolutionary computational technique like PSO is implemented which reduces the search space. Experimental results show that the PSO gives better performance over traditional exhaustive search in the case of fractal image compression. Normally the PSNR ratio for a decoded image should be very high to have a better image and the encoding time should be less. Based on Tab. 2 it can be seen that even though the PSNR value obtained through PSO is less than that of exhaustive search method, the encoding time of PSO based FIC is less than that of traditional method.

References

[1] M. Barnsley, A. Jacquin. Application of recurrent iterated function system to images. **in:** *Proceedings of SPIE*, *3*, vol. 1001, 1998, 122–131.
[2] Y. Chakrapani, K. Rajan. A comparative approach to fractal image compression using genetic algorithm and simulated annealing technique. *Asian Journal of Information technology*, **7**(7).
[3] D. Duh, J. Jeng, S. Chen. DCT based simple classification scheme for fractal image compression. *Image and Vision Computing*, 2005, **23**(13): 1115–1121.
[4] A. Jacquin. Image coding based on a fractal theory of iterated contractive image transformation. *IEEE Transactions on Image Processing*, 1992, **1**: 18–30.
[5] A. Jacquin. Fractal image coding a review. **in:** *Proceedings of IEEE*, *10*, vol. 81, 1993, 212–223.
[6] J. Kennedy, R. Eberhart. Particle swarm optimization. **in:** *Proceedings of IEEE International Conference on Neural Networks(ICNN'95)*, vol. 4, Perth, Australia, 1995, 1942–1948.
[7] J. Kennedy, R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, CA, 2001.
[8] K. Lee, M. El-Sharkawi. *Modern HeuristicOptimization Techniques with applications to Power Systems*. IEEE Power Engineering Society (02TP160), 2002.
[9] M. Li, S. Ou, H. Zhang. The new progress in research approach of fractal image compression. *Journal of Engineering Graphics*, 2004, **4**(3): 143–152.
[10] J. Park, K. Lee, et al. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power Systems*, 2005, **20**(1): 34–42.