VOL. 4, NO. 1, FEBRUARY 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

GENETIC ALGORITHM APPLIED TO FRACTAL IMAGE COMPRESSION

Y. Chakrapani[1] and K. Soundara Rajan[2]
[1]Department of Electronics and Communications, G. Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India
[2]Department of Electronics and Communications, J.N.T.U College of Engineering, Anantapur, Andhra Pradesh, India
E-Mail: yerramsettychakri@gmail.com

ABSTRACT

In this paper the technique of Genetic Algorithm (GA) 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, Genetic algorithm is proposed. Experimental results show that the Genetic Algorithm is a better method than the traditional exhaustive search method.

Keywords: fractal image compression, genetic algorithm, crossover, mutation, 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] 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 [5]. Especially in the field of reducing the complexity of search, many outstanding algorithms based on classified search have been proposed.

GA is a search and optimisation method developed by mimicking the evolutionary principles and chromosomal processing in natural genetics. GAs are general purpose optimization techniques based on principles inspired from the biological evolution using metaphors of mechanisms such as natural selection, genetic recombination and survival of the fittest. They are member of a wider population of algorithm, Evolutionary Algorithm (EA). The idea of evolutionary computing was introduced in the year 1960 by I. Rechenberg in his work “evolution strategies” (“Evolutionsstrategie” in original). His idea was then developed by other researchers. Genetic Algorithm (GA) was invented by John Holland and Thereafter numbers of his students and other researchers have contributed in developing this field. With the advent of the GA, many non-linear, large-scale combinatorial optimization problems in power systems have been resolved using the genetic computing scheme.

The GA is a stochastic search or optimization procedure based on the mechanics of natural selection and natural genetics. The GA requires only a binary representation of the decision variables to perform the genetic operations, i.e., selection; crossover and mutation. Especially GA is efficient to solve nonlinear multiple-extrema problems [6-8] and is usually applied to optimize controlled parameters and constrained functions.

A few investigations have been carried out in application of GA to fractal image compression. However these authors have not explained a clear way of applying evolutionary computational techniques like GA to fractal image compression and hence their work does not give a clear idea about the working of evolutionary techniques in 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 GA and its application to FIC is explained. section 4 and 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 $W(x) = T(x) + b$, where T is a linear transformation

on R^n and $b \in R^n$ 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

$d(W(x), W(y)) \le cd(x, y)$ 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 $\{w_1, ...w_m\}$ of contractions is known as Local Iterated function scheme (LIFS). If there is a subset $F \subseteq D$ Such that for a LIFS $\{w_1, ...w_m\}$

$$F = \bigcup_{i=1}^{m} w_i(F) \quad (1)$$

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 S denote the class of all non-empty compact subsets of D. The δ-parallel body of $A \in S$ is the set of points within distance δ of A, i.e.

$$A_\delta = \{x \in D : |x - a| \le \delta, a \in A\} \quad (2)$$

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 $\{w_1, ...w_m\}$, 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

$$W(E) = \bigcup_{i=1}^{m} w_i(F) \quad (3)$$

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 $K \ge 1$ then we have that

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

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.

(a) Collage Theorem

Let $\{w_1, ...w_m\}$ 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

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

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 $\{w_1, ...w_m\}$ with invariant set F satisfying $d(E, F) < \delta$. Therefore the problem of finding a LIFS $\{w_1, ...w_m\}$ 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.

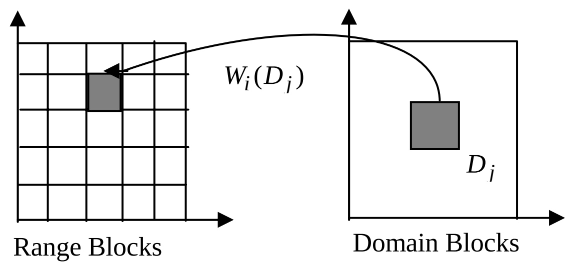


Figure-1. Domain- Range block transformations.

VOL. 4, NO. 1, FEBRUARY 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

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. The basic algorithm for fractal encoding is as follows:

a) 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.
b) 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.
c) 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 Figure-1.
d) 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 map onto that range cell and parameters that describe the transformation mapping the domain onto the range.

The above fractal image coding algorithm can be represented in the form of flowchart as shown in Figure 8. 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.

3. GENETIC ALGORITHM

Genetic algorithms are procedures based on the principles of natural selection and natural genetics that have proved to be very efficient in searching for approximations to global optima in large and complex spaces in relatively short time. The basic components of GA are:

- Representation of problem to be solved;
- Genetic operators (selection, crossover, mutation);
- Fitness function; and
- Initialization procedure.

3.1 Biological background of GA

All living organisms consist of number of cells. Each cell consists of same set of chromosomes. Chromosomes are strings of DNA and serves as a model for the whole organism. A chromosome's characteristic is determined by the genes. Each gene has several forms or alternatives which are called alleles, producing differences in the set of characteristics associated with that gene.

The set of chromosome is called the genotype, which defines a phenotype (individual) with certain fitness. The fitness of an organism is measured by success of the organism in its life. According to Darwinian theory the highly fit individuals are given opportunities to reproduce whereas the least fit members of the population are less likely to get selected for reproduction and so "die out".

GA starts by using the initialization procedure to generate the first population. The members of the population are usually strings of symbols (chromosomes) that represent possible solutions to the problem to be solved as shown in Figure-2.

$$\underbrace{1\,1\,0\,1\,0}_{x_1}\;\underbrace{1\,0\,0\,1\,0\,0\,1}_{x_2}\;\underbrace{0\,1\,0}_{x_3}\;\text{------}\;\underbrace{0\,0\,1\,0}_{x_n}$$

Figure-2. String representing 'N' variables.

Each of the members of the population for the given generation is evaluated and according to its fitness value, it is assigned a probability to be selected for reproduction. Using this probability distribution, the genetic operators select some of the individuals. By applying the operators to them, new individuals are obtained. The mating operator selects two members of the population and combines their respective chromosomes to create offspring as shown in Figure-3. The mutation operator selects a member of the population and changes part of the chromosome as shown in Figure 4.

Parent A	0 0 0 0 0 1 0 1
Parent B	1 1 1 0 1 0 0 1
Child A	0 0 1 0 1 0 0 1
Child B	1 1 0 0 0 1 0 1

Figure-3. Binary crossover.

Child A 0 1 0 1 1

↓

New Child A 0 1 1 1 1

Figure-4. Binary mutation.

The algorithm for the genetic algorithm can be represented as follows.

Step 1: As the genetic algorithm takes pairs of strings, we create a random number of strings depending upon our necessity and also note down their decoded values along with setting a maximum allowable generation number tmax.
Step 2: Using the mapping rule we next find out the corresponding values of above created strings.
Step 3: Using these values the fitness function values are found out.
Step 4: Next the process of reproduction is carried out on the strings to create a mating pool.
Step 5: The process of crossover and mutation is also

carried out on the strings with probabilities of 0.8 and 0.05, respectively.

Step 6: After the termination criteria is met with, the value of string with minimum fitness function value is considered as optimum value.

The various parameters of Genetic Algorithm are shown in Table-1.

4. SYSTEM INVESTIGATION

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. In the case of Genetic Algorithm the individual chromosome is coded as shown in Figure-5.

$$\underbrace{1\,1\,0\,1\,0\,1\,0\,1}_{x}\;\underbrace{1\,0\,0\,1\,0\,0\,1\,1}_{y}\;\underbrace{0\,1\,0}$$

Domain block coordinates Isometry type

Figure-5. Binary code for searching domain block.

The mean squared error (MSE) and PSNR considered in this work are given by

$$MSE=\frac{1}{N_{Rows}N_{Cols}}\sum_{i=1}^{N_{Rows}}\sum_{j=1}^{N_{Cols}}\left|f_{i,j}-d_{i,j}\right|^2 \qquad (6)$$

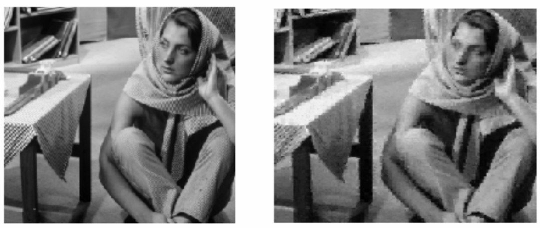
$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \qquad (7)$$

The genetic operators are applied to the string having more fitness value.

5. RESULTS AND DISCUSSIONS

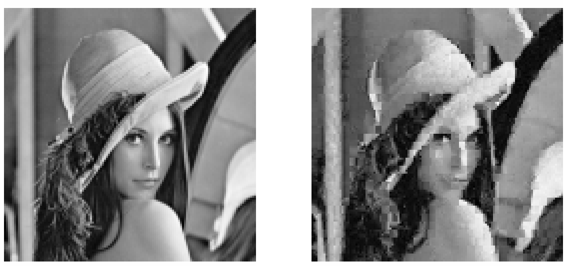
This work is carried out in MATLAB 7.0 version on Pentium-4 processor with 1.73 GHz and 256 MB RAM and the original image is classical 128×128 Lena and Barbara face image coded with 8 bits per pixel. An optimal bit allocation strategy for GA is as follows: 14 bits for the location of matched domain block (horizontal and vertical coordinate), 3 bits for isomorphic types. For each of the range block fractal coding includes 17 bits allocation.

During the iteration process of the above proposed methods the gray level values beyond 0 and 255 are replaced by average of its four neighbours to avoid block diverging. Figures 6 and 7 shows the reconstructed images using FIC with GA as search algorithm along with the original images of Barbara and Lena after 10 iterations. The Coding scheme of FIC using GA is given in Table-2.



(a) (B)

Figure-6. (a) Original image. (b) Reconstructed image using GA.



(a) (B)

Figure-7. (a) Original image. (b) Reconstructed image using GA.

VOL. 4, NO. 1, FEBRUARY 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

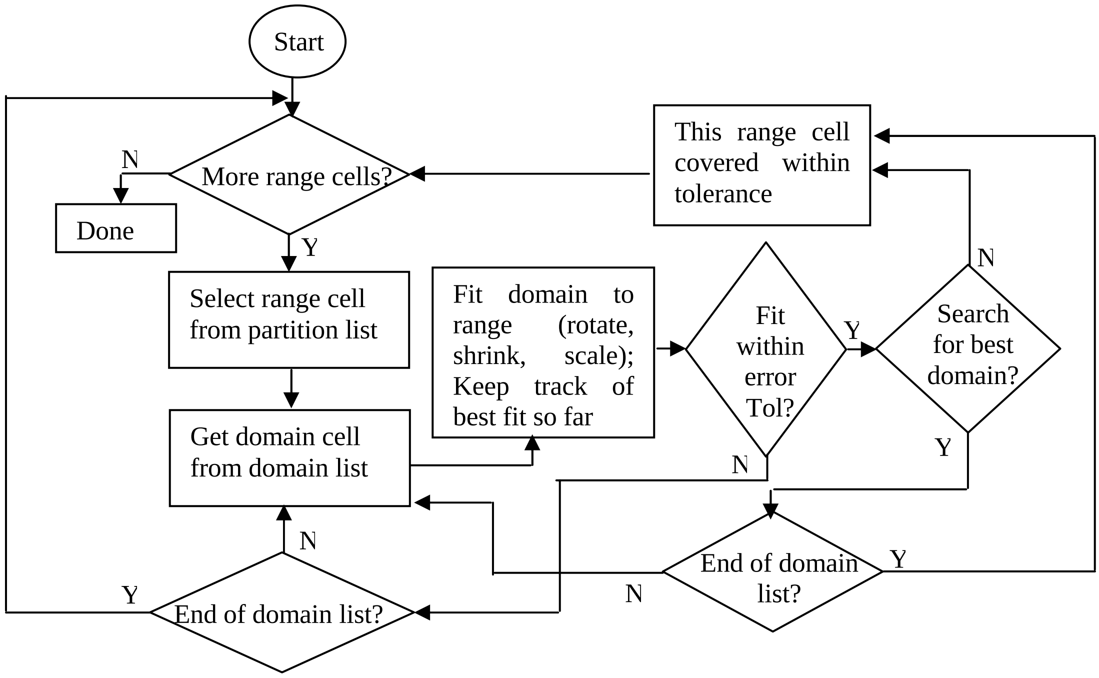


Figure-8. Flow chart of fractal image encoding.

Table-1. Genetic Algorithm's parameters.

Range block size	**4*4**	**Population size**	**50**
Crossover probability Pc	0.8	Fitness f	1/(1+MSE)
Mutation probability	0.05	Iterations	10

Table-2. Coding scheme comparison with GA and SA.

	Image	**FIC with exhaustive search**	**FIC with Genetic Algorithm**
Compression ratio	Barbara	1.2:1	6.73:1
	Lena	1.3:1	6.73:1
PSNR (db)	Barbara	32.84	28.34
	Lena	32.69	26.22
Encoding time (sec)	Barbara	8400	2500
	Lena	8400	2370

6. CONCLUSIONS

In this paper the concept of GA is applied to FIC. Instead of global searching in FIC the evolutionary computational technique like GA is implemented which shortens the search space. Experimental results show that the GA 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. Based on Table-2 it can be seen that the PSNR and Compression ratio are better in the case of decoded image using GA over the one obtained by exhaustive search technique. The performance of GA can be further improved by introducing the concept of elitism which copies the best string in one generation into the second generation.

REFERENCES

[1] A.E. Jacquin. 1992. Image coding Based on a Fractal theory of Iterated contractive Image Transformations. IEEE Transactions on Image Processing. Vol. 1. pp. 18-30.

[2] A.E. Jacquin. 1993. Fractal Image coding: A Review Proc. IEEE. Vol. 81. pp. 1451-1465.

[3] M.F. Barnsley. 1988. Fractals Everywhere. New York: Academic.

[4] M.F Barnsley and A.E Jacquin. 1998. Application of recurrent iterative function systems to images. Proc. SPIE. 1001(3): 122-131.

[5] Mingshui Li, Shanhu Ou and Heng Zhang. 2004. The new progress in research approach of fractal image compression. Journal of Engineering Graphics. 4(3): 143-152.

[6] Xiaoping Wang, Liming Cao. 2002. Genetic Algorithms-theory, application and software reliability. Xi a Jiao Tong University Press.

[7] J.H. Holland. 1975. Adaptation in nature and artificial Systems. University of Michigan press: Michigan.

[8] David Goldberg. 1989. Genetic algorithms in search, optimization and machine learning Addison-Wesley Publishing Company: MA.