# Correspondence

## Technique for Fractal Image Compression Using Genetic Algorithm

Suman K. Mitra, C. A. Murthy, and Malay K. Kundu

***Abstract*—A new method for fractal image compression is proposed using genetic algorithm (GA) with elitist model. The self transformability property of images is assumed and exploited in the fractal image compression technique. The technique described here utilizes the GA, which greatly decreases the search space for finding the self similarities in the given image. This correspondence presents theory, implementation, and analytical study of the proposed method along with a simple classification scheme. Comparison with other fractal-based image compression methods has also been reported here.**

***Index Terms*—Compression ratio, genetic algorithm (GA), image compression, isometry, iterated function system (IFS).**

### I. Introduction

The theory of image coding using iterative function system (IFS) was first proposed by Barnsley [1]. He modeled real life images by means of deterministic fractal objects, i.e., by the attractors evolved through iterations of a set of contractive affine transformations. With the help of iterated function system, along with collage theorem, Barnsley laid the foundation of the fractal-based image compression. A set of contractive affine transformations—iterative function system (IFS)—can approximate a real image and so, instead of storing the whole image, it is enough to store the relevant parameters of the transformations reducing memory requirements. The basic problem of fractal-based image compression is to find appropriate parameter values of transformations whose attractor is an approximation of the given image. A review on fractal-based image coding methodology is available in the literature [2].

A fully automated fractal-based image compression technique of digital monochrome image was first proposed by Jacquin [3]. The encoding process consists of approximating the small image blocks, called range blocks, from the larger blocks, called domain blocks, of the image, through some operations. In the encoding process, separate transformations for each range block are obtained. The scheme also uses the theory of vector quantization [4] to classify the blocks. The set consisting of these transformations, when iterated upon any initial image, will produce a fixed point (attractor) that approximates the target image. This scheme can be viewed as partitioned iterative function system (PIFS). One such scheme, using PIFS, to store fewer number of bits (or to increase the compression ratio) was proposed by Fisher *et al.* [5].

Genetic algorithms (GA's) [6]–[12] are mathematically motivated search techniques that try to emulate biological evolutionary processes to solve optimization problems. Instead of searching one point at a time, GA's use multiple search points. GA's attempt to find near-optimal solutions without going through an exhaustive search mechanism. Thus, GA's can claim significant advantage of large reduction in search space and time.

In the present work, a new method for image compression using PIFS is proposed. The proposed method uses a simpler classification system for range blocks. Genetic algorithms with elitist model are used in finding the appropriate domain block as well as the appropriate transformation for each range block. An analytical study of the proposed method along with a comparison with other existing methods is also reported here.

Theory and key features of IFS and GA's are outlined in Section II. The method of selecting fractal codes for a given image using GA's is described in Section III. Section IV presents implementation and results. In Section V analytical study of the proposed method and comparison with the other fractal-based methods are described followed by discussion and conclusions in Section VI.

### II. Theory and Basic Principles

#### *A. Theoretical Foundation of Image Coding by IFS*

The detailed mathematical descriptions of the IFS theory, collage theorem, and other relevant results are available in [1], [13], [14]. Only the salient features of image coding through IFS are given below.

Let $I$ be a given image that belongs to the set $X$. Generally, $X$ is taken as the collection of compact sets. Our intention is to find a set $\mathcal{F}$ of affine contractive maps for which the given image $I$ is an approximate fixed point. $\mathcal{F}$ is constructed in such a way that the distance between the given image and the fixed point (attractor) of $\mathcal{F}$ is very small. The attractor "$A$" of the set of maps $\mathcal{F}$ is defined as follows:

$$\lim_{N\to\infty} \mathcal{F}^N(J) = A, \qquad \forall J \in X$$

and $\mathcal{F}(A) = A$, where $\mathcal{F}^N(J)$ is defined as

$$\mathcal{F}^N(J) = \mathcal{F}[\mathcal{F}^{N-1}(J)],$$

with

$$\mathcal{F}^1(J) = \mathcal{F}(J),\ \forall J \in X.$$

Also, the set of maps $\mathcal{F}$ is defined as follows:

$$d[\mathcal{F}(J_1), \mathcal{F}(J_2)] \le s\, d(J_1, J_2); \quad \forall J_1, J_2 \in X \text{ and } 0 \le s < 1. \tag{1}$$

Here, $d$ is called the distance measure and $s$ is called the contractivity factor of $\mathcal{F}$: Let

$$d[I, \mathcal{F}(I)] \le \epsilon \tag{2}$$

where $\epsilon$ is a small positive quantity. Now, by the collage theorem [1], it can be shown that

$$d(I, A) \le \frac{\epsilon}{1-s} \tag{3}$$

where $A$ is the attractor of $\mathcal{F}$.

From (3) it is clear that, after a sufficiently large number ($N$) of iterations, the set of affine contractive maps $\mathcal{F}$ produces a set that

Manuscript received December 1, 1995; revised May 14, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael W. Marcellin.

The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700035 India (e-mail: res9432@isical.ernet.in; murthy@isical.ernet.in; malay@isical.ernet.in).

Publisher Item Identifier S 1057-7149(98)02464-6.

belongs to $X$ and is very close to the given original image $I$. Here, $(X, \mathcal{F})$ is called iterative function system and $\mathcal{F}$ is called the set of fractal codes for the given image $I$.

### B. Basic Principles and Features of Genetic Algorithms

GA's are adaptive search processes based on the notion of selection mechanism of natural genetic system [6]. GA's help to find the global near optimal solution without getting stuck at local optima as they deal with multiple points (spread all over the search space) simultaneously. To solve the optimization problem, the GA starts with the structural representation of a parameter set. The parameter set is coded as a string of finite length and the string is called chromosome. Usually, the chromosomes are strings of zero's and one's. If the length of string (chromosome) is $l$, then the total number of possible strings is $2^l$.

*1) Fitness Function:* Usually, a function, called *fitness function* is defined on the set of chromosomes (strings). The problem here is to find the string (chromosome) that provides optimal fitness value among all strings (chromosomes). In this work, we are dealing with a minimization problem. The definition of fitness function is provided in Section III-C. ♠

*2) Initial Population:* Out of all possible $2^l$ strings, initially a few strings (say $S$ number of strings) are selected randomly and this set of strings is called initial population [6]. In this article, we have taken $S$ to be an even number. ♠

Three basic genetic operators, i) selection, ii) crossover, and iii) mutation, are exploited in GA's. The genetic operators are applied on the initial population to give rise to a new population of same size ($S$). The operators are again applied on this new population to give rise to another population. The process of creation of a new population from the existing one is called iteration. In this article the process is executed for a fixed number of iterations. It is proved that the elitist model of GA will find the optimal solution as the number of iterations tend to infinity [15]. Usually, for real-life problems, satisfactory stopping times of the GA are found after several experiments. For the present problem too, the stopping time is found after several experiments. The descriptions of the different operations are given below.

*3) Selection:* In this operation, a mating pool of strings of the current population is generated by using the fitness values of strings in the population. The probability of selection of a string in the population to the mating pool is inversely proportional to its fitness value. (Note that the present optimization problem is a minimization problem.) This scheme is known as proportional selection scheme. This scheme provides more representatives of the string having optimal fitness value among all the strings in the present population. The size or the number of strings in a mating pool is same as that of the initial population. ♠

*4) Crossover:* There are several ways of performing the crossover operation [7] among the strings in the mating pool. The single-point crossover operation is followed in this article. The crossover probability is represented by $P_{\text{cross}}$.

In this operation, $S/2$ pairs of strings are formed randomly from the mating pool, of size $S$. A random number $rand$ in the range [0, 1] is then generated for each pair of strings. If $rand \leq P_{\text{cross}}$; then crossover is performed. Otherwise, the operation is not performed. Each pair of strings undergoes crossing over in the following manner. An integer position $k$ is selected randomly between 1 and $l-1$ ($l \geq 2$, is the string length). Two new strings are then created by swapping all the characters from position $k+1$ to $l$ of the old strings. Usually a high value is assigned for the crossover probability $P_{\text{cross}}$. ♠

*5) Mutation:* In this operation every bit of every string is flipped (i.e., zero by one and one by zero) with probability $P_{mut}$. $P_{mut}$ is called the *mutation probability*.

One of the commonly used conventions is to assign a very small value to the mutation probability $P_{mut}$ and keep the $P_{mut}$ fixed for all the iterations, i.e., the value for $P_{mut}$ is independent of the number of iterations. A different strategy is adopted here in assigning the value for $P_{mut}$. We have prefixed the number of iterations of GA *a priori* and varied the mutation probability with the number of iterations. This varying mutation probability scheme has already been applied successfully in connection with an application of GA's to pattern recognition problem [16]. ♠

*6) Elitist Model:* In the elitist model [8] of GA's, the knowledge about the best string obtained so far is usually preserved within the population. For this purpose, the worst string in the present population is replaced by the best string of the previous population in each iteration. The best strings of two consecutive iterations will be always in the mating pool of the present population in elitist model. Thus, this model will provide a track of best strings in all the iterations. ♠

In the next section, the methodology of construction of fractal code using GA's is described.

## III. Proposed Methodology

### A. Construction of Fractal Codes

Let, $I$ be a given image having size $w \times w$ and the range of gray-level values be $[0, g]$. Thus, the given image $I$ is a subset of $\mathbb{R}^3$. The image is partitioned into $n$ nonoverlapping squares of size, say $b \times b$, and let this partition be represented by $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is named as range block. Note that $n = w/b \times w/b$. Let $\mathcal{D}$ be the collection of all possible blocks which is of size $2b \times 2b$ and let $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Each $\mathcal{D}_j$ is named as domain block with $m = (w - 2b) \times (w - 2b)$.

Let

$$\mathcal{F}_j = \{f: \mathcal{D}_j \to \mathbb{R}^3;\ f \text{ is an affine contractive map}\}.$$

Now, for a given range block $\mathcal{R}_i$, let, $f_{i|j} \in \mathcal{F}_j$ be such that

$$d[\mathcal{R}_i,\, f_{i|j}(\mathcal{D}_j)] \leq d[\mathcal{R}_i,\, f(\mathcal{D}_j)] \qquad \forall f \in \mathcal{F}_j, \forall j.$$

Now let $k$ be such that

$$d[\mathcal{R}_i,\, f_{i|k}(\mathcal{D}_k)] = \min_j \{d[\mathcal{R}_i,\, f_{i|j}(\mathcal{D}_j)]\}. \tag{4}$$

Also, let $f_{i|k}(\mathcal{D}_k) = \widehat{\mathcal{R}}_{i|k}$.

Our aim is to find $f_{i|k}(\mathcal{D}_k)$ for each $i \in \{1, 2, \cdots, n\}$. The set of maps $\mathcal{F} = \{f_{1|\bullet}, f_{2|\bullet}, \cdots, f_{n|\bullet}\}$ thus obtained is called the fractal code of image $I$. Also note that there are other works on fractal image compression in which the domain and range blocks are utilized [3], [5].

To find the best matched domain block as well as the best matched map, one has to search all possible domain blocks with the help of (4). The affine contractive map $f_{i|\bullet}$ is constructed in two steps. The first step is transformation of rows and columns from domain blocks to range blocks. This part is nothing but the change of coordinates in a two-dimensional (2-D) geometry. This can be achieved by using any one of the eight possible transformations, called *isometries*, on the domain blocks [3]. Once the first part is obtained, second part is estimation of a set of pixel values of range blocks from the set of pixel values of the transformed domain blocks. These estimates can be obtained by using the least square analysis of two sets of pixel values.

The distance measure "$d$" [used in (4)] is taken to be the simple mean square error (MSE) between the original set of gray values and the obtained set of gray values of the concerned range block, viz, $\mathcal{R}_i(p, q)$ and $\widehat{\mathcal{R}}_{i|k}(p, q)$, respectively. As selection of fractal code for a range block is dependent only on the estimation of pixel values of that block, it is enough to calculate only the distortion of the original and estimated pixel values of the block. Thus, the MSE is taken as the distance measure. Note that the same measure had also been used in other works [3], [5]. The fractal code $\mathcal{F}$ is used for decoding the given image $I$ from any arbitrary starting image $I_0$.

A two-level image partition scheme as reported by Jacquin [3] is used for the implementation of the GA-based fractal image compression scheme.

The GA is used to search for an appropriately matched domain block as well as an appropriate transformation (isometry) for a particular type of range block. The class of range blocks is obtained through a simple classification scheme.

### B. Classification

The purpose of block classification is twofold. One purpose is to store fewer number of bits or to get higher compression ratio and the other is to reduce the encoding time. A simple classification scheme on range blocks alone is used here. Range blocks are grouped into two sets according to the variability of the pixel values in these blocks. If the variability of a block is low, i.e., if the variance of the pixel values in the block is below a fixed value, called the *threshold*, we call the block as smooth type range block. Otherwise, we call it a rough type range block. The threshold value that separates the range blocks into two types is obtained from the valley in the histogram of the variances of pixel values within blocks. After classification, GA-based encoding is adopted for rough type range blocks. All the pixel values in a smooth type range block are replaced by the mean of its pixel values. The scheme mentioned above is a time saving one provided the number of smooth type range blocks is significant. The analysis of the proposed method (Section V) discusses these aspects.

### C. GA to Find Fractal Codes

The main aspect of fractal-based image coding is to find a suitable domain block and a transformation for a rough type range block. Thus, the whole problem can be looked upon as a search problem. Instead of a global search mechanism we have introduced GA's to find the near optimal solution.

The number of possible domain blocks to be searched are $(w - 2b) \times (w - 2b)$ and the number of transformations to be searched for each domain block is eight (Section III-A). Thus, the space to be searched consists of $M$ elements. $M$ is called cardinality of the search space. Here, $M = 8(w - 2b)^2$. Let the space to be searched be represented by $\mathcal{P}$ where

$$\mathcal{P} = \{1, 2, \cdots, (w - 2b)\} \times \{1, 2, \cdots, (w - 2b)\} \times \{1, 2, \cdots, 8\}.$$

Binary strings are introduced to represent the elements of $\mathcal{P}$. The set of $2^l$ binary strings, each of length $l$, are constructed in such a way that the set exhausts the whole parametric space. The value for $l$ depends on the values of $w$ and $b$. The fitness value of a string is taken to be the MSE between the given range block and the obtained range block.

Let $S$ be the population size and $T$ be the maximum number of iterations for the GA. Note that the total number of strings searched up to $T$ iterations is $S \times T$. Hence, $M/ST$ provides the search space reduction ratio for each rough type range block.

## IV. Implementation and Results

The GA based method discussed in Section III is implemented in 256 × 256, 8 b/pixel Lena image. The image is subdivided into four, 128 × 128 subimages, each of which is encoded separately.

For the specific implementation of the classification scheme mentioned above, the variances of pixel values of all 8 × 8 and 4 × 4 range blocks are computed and corresponding thresholds are selected from the respective histograms of the variances. For 8 × 8 range blocks, a valley is found near the value 20 in the histogram and thus, this value is chosen as threshold for 8 × 8 range blocks. Similarly, 35 is taken as the threshold value for the 4 × 4 range blocks.

TABLE I
Test Results for 256 × 256, 8 B/Pixel Lena Image

| Type of encoding | Range block size | Domain block size | Number of range blocks: Parent | | Child | | Compression ratio | Bits per pixel | PSNR (in db) |
|---|---|---|---|---|---|---|---|---|---|
| | | | Smooth | Rough | Smooth | Rough | | | |
| Single level | 8 × 8 | 16 × 16 | 278 | 746 | Nil | Nil | 21.7 | 0.37 | 26.16 |
| Two level | 8 × 8 and 4 × 4 | 16 × 16 and 8 × 8 | 278 | 533 | 128 | 951 | 10.50 | 0.76 | 30.22 |

Considering parent range blocks of size 8 × 8 and children range blocks of size 4 × 4 and using two level image partition scheme [3] each subimage is then encoded. The methodology proposed in this article is also implemented with a single level partition scheme where only parent blocks are considered.

GA's are implemented, as a search technique, only for rough type range blocks. Here for each subimage, total number of parent range blocks is $n = 256$ and total number of domain blocks ($m$) to search is (128 − 16) × (128 − 16) = 112 × 112 and (128 − 8) × (128 − 8) = 120 × 120 for parent and child range blocks, respectively. Thus, the cardinality ($M$) of the search spaces for these two cases are 112 × 112 × 8 and 120 × 120 × 8, respectively. The string length $l$ has been taken to be $17(7 + 7 + 3)$ in both the cases. As a result of selecting $2^{17}$ binary strings, a few strings, in both the cases, will be outside the specified search spaces. If a string which is not in the search space is selected during the implementation of the GA, then a string at the boundary will replace it.

Out of these $2^{17}$ binary strings, six strings $(S = 6)$ are selected randomly to construct an initial population. A high probability, say $P_{\text{cross}} = 0.85$, is taken for the crossover operation. For mutation operation, $P_{mut}$ (mutation probability) is varied over the iterations and the exact values are 0.30, 0.20, 0.15, 0.10, and 0.06. The total number of iterations considered in the GA is $T = 910$. Hence, the search space reduction ratios for a parent and a child rough type range blocks are approximately 18 and 21, respectively.

For encoding the Lena image, the results of both one level (i.e., only 8 × 8 range blocks) and two level (i.e., first 8 × 8 and then 4 × 4) encoding are reported in this article. Test results and some statistics of both the cases are given in Table I.

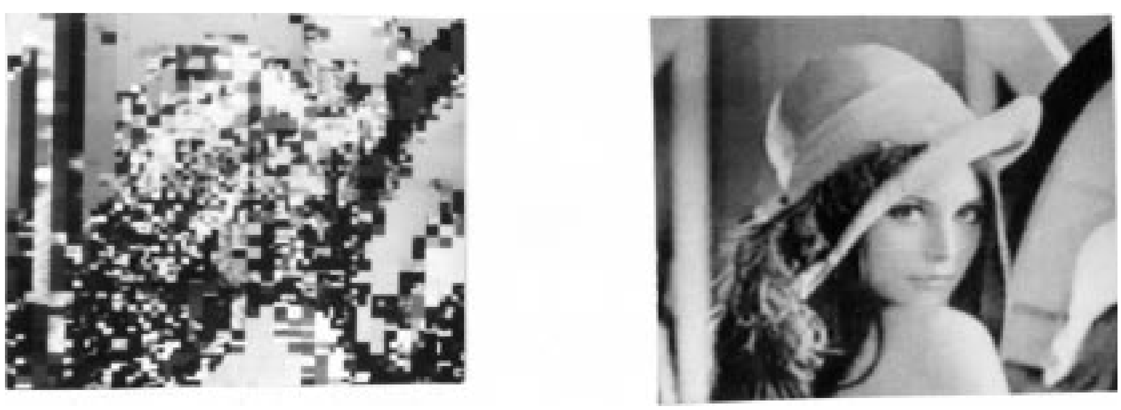
The corresponding diagrams for the Lena image are shown in Figs. 1–3. Fig. 1(a) shows the original Lena image, 1(b) shows an arbitrary starting image of Rose, and Fig. 1(c) and (d) are the decoded Lena image (after ten iterations) using single level and two level encodings, respectively. The intermediate results of the decoding process have been provided in Figs. 2 and 3. Fig. 2(a) and (b) are results obtained in iteration one and iteration ten, respectively, while decoding the Lena image starting from the Rose image. Fig. 3(a) and (b) are, respectively, the difference images of decoded Lena image from the original Lena image after iteration one and iteration ten.

The algorithm is also tested for the girl image and the low-flying aircraft (LFA) image. Both these images are 8 b/pixel images of size 256 × 256. Fig. 4(a) and (b) are original and decoded (after ten iterations) girl images, respectively, with compression ratio 11.37 and peak signal-to-noise ratio (PSNR) 30.74. Fig. 5(a) and (b) are original and decoded (after ten iterations) LFA images, respectively, with compression ratio 5.51 and PSNR 26.86. Note that the search space reduction ratios for both these images are same as that of Lena

(a) (b)

(d) (c)

Fig. 1. (a) Original Lena image. (b) Rose image. (c) Decoded Lena image after ten iterations using single level with classification scheme with the starting image as Fig. 1(b). (d) Decoded Lena image after ten iterations using two level with classification scheme with the starting image as Fig. 1(b).



(a) (b)

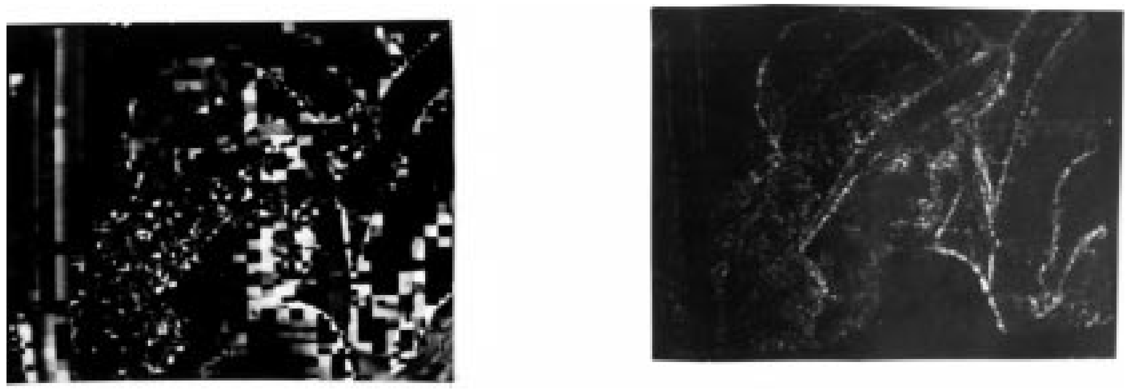
Fig. 2. (a) Result of decoded Lena image after iteration number one. (b) Result of decoded Lena image after iteration number ten.

image since sizes of parent and child range blocks as well as $S$ and $T$ are same.

The GA based technique of fractal image compression method is also compared with exhaustive search mechanism. The test results of single level ($8 \times 8$ range block size) and two level ($8 \times 8$ and $4 \times 4$ range block size) encoding scheme of both the techniques are shown in Table II. Note that all the prefixed parameters are same in both cases.

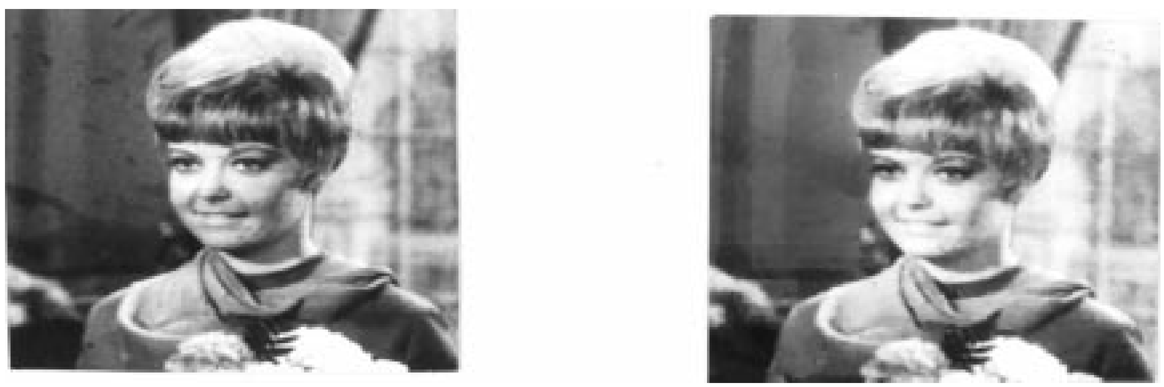
It is very clear from the test results that for single level encoding the compression ratios are found to be same in both techniques. On the other hand, the PSNR values are very close to each other. But the number of domain blocks searched in both cases provides the justification of using the GA as a search technique. The number of domain blocks searched in case of GA is at least 20 times smaller than that of in the case of exhaustive search mechanism.

In two level encoding scheme, more compression is achieved in exhaustive search. In this encoding scheme, in the first level, more rough type parent range blocks are encoded correctly (in the sense of error threshold) in exhaustive search technique. These blocks are not divided into child blocks for second level encoding. As a result of this, more compression is achieved in exhaustive search case. But the PSNR value appears to be better in case of the GA-based



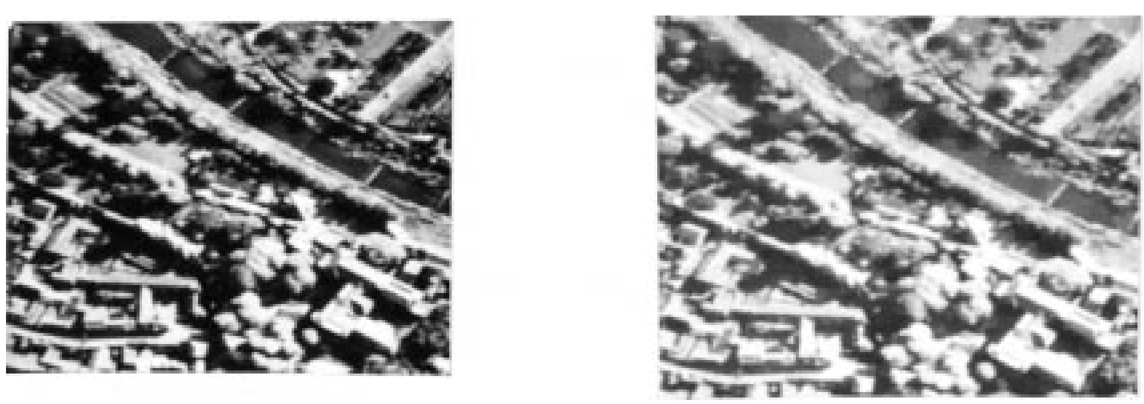
(a) (b)

Fig. 3. (a) Difference image of decoded Lena from original after iteration number one. (b) Difference image of decoded Lena from original after iteration number ten.



(a) (b)

Fig. 4. (a) Original girl image. (b) Decoded girl image after ten iterations using two level with classification scheme with the starting image as Fig. 1(b).



(a) (b)

Fig. 5. (a) Original LFA image. (b) Decoded LFA image after ten iterations using two level with classification scheme with the starting image as Fig. 1(b).

technique. Moreover, the advantage of using the GA-based technique is established from the value of the number of domain blocks searched in both the cases. Here also, the GA-based technique searched at least 20 times fewer domain blocks in comparison with the exhaustive search technique.

The search space reduction is achieved, since near-optimal solutions are usually satisfactory and, intuitively, the solutions whose fitness values are far away from the optimal are thrown away in a bulk. This is the reason GA performs well for optimization problems [17].

## V. Comparision and Analysis

### A. Comparison

Advantages of GA-based fractal image coding are clearly demonstrated in reducing the search space for finding the appropriately matched domain block and the appropriate transformation corresponding to a range block. The GA based method is found to provide computational efficiency, thereby drastically reducing the cost of coding. The obtained results for the Lena image are also

TABLE II
RESULTS OBTAINED BY USING THE GA-BASED TECHNIQUE AND THE EXHAUSTIVE SEARCH TECHNIQUE FOR LENA IMAGE

| Type of encoding | Genetic Algorithm | | | Exhaustive Search | | |
|---|---|---|---|---|---|---|
| | Compression Ratio | PSNR in db | Number of domain block searched | Compression Ratio | PSNR in db | Number of domain block searched |
| Single level $8 \times 8$ | 21.7 | 26.16 | 4073160 | 21.7 | 26.20 | 85939200 |
| Two level $8 \times 8$ & $4 \times 4$ | 1 0.50 | 30.22 | 8102640 | 11.24 | 28.32 | 161869952 |

TABLE III
NUMBER OF BITS TO BE STORED IN DIFFERENT SCHEMES OF A $256 \times 256$, 8 B/PIXEL IMAGE DURING THE DECODING PROCESS

| | | Single level $(8 \times 8)$ | Two level $((8 \times 8)$ & $(4 \times 4))$ | | |
|---|---|---|---|---|---|
| | | | All parent | All child | Mixed (parent & child) |
| Without classification | | $28 \times 1024$ | $28 \times 1024 + 1024 \times 4$ | $28 \times 4096 + 1024 \times 4$ | $28 \times c + 1024 \times 4$ |
| With classification | All smooth | $9 \times 1024$ | $9 \times 1024 + 1024 \times 4$ | $9 \times 4096 + 1024 \times 4$ | $9 \times c + 1024 \times 4$ |
| | All rough | $29 \times 1024$ | $29 \times 1024 + 1024 \times 4$ | $29 \times 4096 + 1024 \times 4$ | $29 \times c + 1024 \times 4$ |
| | Mixed (smooth & rough) | $29 \times 1024 - 20 \times r$ | $29 \times 1024 - 20 \times r + 1024 \times 4$ | $29 \times 4096 - 20 \times r + 1024 \times 4$ | $29 \times c - 20 \times r + 1024 \times 4$ |

comparable with that of some of the existing methods of fractal image compression [3], [5].

A different kind of trimming of the search space was described by Jacquin [3]. In his method reduction takes place in two steps, by making a "domain pool" consisting of domain blocks and then by "classifying" this pool into some classes based on the geometric features of the blocks.

In Jacquin's method starting from the first pixel of the given image, domain blocks are selected by sliding a window of size equal to the size of the domain blocks across the image and taking a constant shift horizontally and vertically. Shift of four pixels and two pixels are considered for selecting the domain pool consisting of domain blocks of size $16 \times 16$ and $8 \times 8$, respectively. Thus the reduction ratios are 16 and 4 for $8 \times 8$ and $4 \times 4$ range blocks, respectively, for a $256 \times 256$ image. On the other hand, the GA-based method reduces the search space corresponding to domain blocks and isometric transformations simultaneously, and the search space reduction ratios for a parent and a child rough type range blocks are approximately 18 and 21, respectively. Moreover, the best matched domain block corresponding to a range block can be located any where within the image support, and so, on trimming the maximal domain pool by shift method, we may lose the best matched domain block in the Jacquin's method. But the GA-based method utilizes the maximal domain pool while searching for the best matched domain block.

The second part of the reduction is obtained using the classification scheme proposed by Ramamurthy and Grasho [4]. This three class classification scheme is adopted to classify both the pool of range blocks as well as domain blocks. This scheme has advantages both in efficient encoding of the range blocks as well as in reducing its search space. But it has limitations, too. According to the scheme, both pools are classified into three classes, so 2 b may be required for storing to indicate the class of the range block under consideration. In the present encoding algorithm, only the range blocks are classified into two classes, which is not only a time-saving scheme, but it also requires storing of only 1 b for class information thereby providing more compression.

GA-based fractal image compression scheme discussed here is also comparable with the algorithm given by Fisher *et al.* [4]. There is a transformation for each range block and the parameters of transformations are stored instead of whole image. Thus, the compression ratio depends on the number of range blocks. More compression can be achieved by considering less number of range blocks. Considering fewer number of range blocks by using quadtree method and $h$–$v$ partitioning method [5], they designed their scheme, which results in the increase of compression ratio. The compression ratio (9.97) and the PSNR (31.53) reported by Fisher *et al.* [5] are almost equal to those reported in this article (Table I).

The search space in the methodology described by Fisher *et al.* [5] is dependent on the complexity of the given image. In particular, in the first step of quadtree method, best domain block for only four range blocks has to be searched. The search process is then carried out up to a fixed level where the minimum size of the range blocks is fixed. The search, in all the intermediate steps, has to be done exhaustively to reach the fixed lowest level in the quadtree method. Thus, an extensive search may need to be carried out for some images. So, in comparison with this method, the proposed GA-based method is better for reducing the search space. Note that, the proposed GA-based (which is a two level scheme) can be extended to a quadtree scheme containing multiple levels, with or without classification of range blocks.

### *B. Analysis*

A two level image partition scheme (parent and its four children) as implemented takes care of finer details of a very small portion of the image. There are altogether 12 possible configurations of four children along with their parents. To indicate the location and presence of child blocks of a parent block, four extra bits for each parent block are needed during encoding. Likewise, an extra bit for each transformation (parent and child) is needed if the classification scheme mentioned in this work is adopted. Thus, the validity of using the classification scheme together with the two level image partition scheme need to be investigated. Comparison of performances of different encoding schemes (single level, two level, without classification, with classification) can be made from the point of view of compression and quality of the decoded image. Table III shows the number of bits required for a $256 \times 256$, 8 b/pixel image under different situations in different schemes. Number of range blocks are 1024 and 4096 with size $8 \times 8$ and $4 \times 4$, respectively. Number of bits require to store each transformation is 28 in the coding scheme not using classification scheme. Coding scheme using classification requires to store 9 b and 29 b, respectively, for the transformation of each smooth type and rough type range blocks.

Table III shows the number of bits to be stored with different schemes for a $256 \times 256$, 8 b/pixel image. The selection of a particular scheme for a given image, which provides suitable values for the PSNR and the compression ratio, depends upon the values of "c" and "r." Here, "c" is the total number of codes in two level partitioning where both parent and child blocks are present and "r" is the number of smooth type range blocks under classification scheme. The following conclusions regarding the usefulness of the application of different schemes during encoding process may be extracted from Table III.

- If a "single level" scheme provides the desired quality of the decoded image (i.e., the desired PSNR), then one should not opt for a "two level" scheme. Note that a two level scheme decreases the compression ratio.

- If all the blocks in a "two level" scheme are of same size, then "single level" encoding with that block size is preferable.
- Under no classification, a "two level" encoding scheme where both parent and child blocks are present is preferable to "single level" scheme if $c < 877$. On the other hand, "single level" encoding with block size 4 × 4 is preferable to this scheme when $c > 3950$. If $c$ lies between 877 and 3950, then switching over from "single level" to "two level" depends upon the quality of the decoded image (i.e., PSNR).

  Under classification, when all the blocks present are "smooth" type, a "two level" encoding scheme where both parent and child blocks are present is preferable to "single level" scheme if $c < 568$. Unlike this, "single level" encoding with block size 4 × 4 is preferable to this scheme when $c > 3641$. If $c$ lies between 568 and 3641, then switching over from "single level" to "two level" depends upon the quality of the decoded image (i.e., PSNR).

  Similarly, with classification, when all the blocks present are "rough" type or "mixed "type (both smooth and rough type blocks are present), a "two level" encoding scheme where both parent and child blocks are present is preferable to "single level" scheme if $c < 880$. On the other hand, "single level" encoding with block size 4 × 4 is preferable to this scheme when $c > 3955$. If $c$ lies between 880 and 3955 then switching over from "single level" to "two level" depends upon the quality of the decoded image (i.e., PSNR).
- Let $r_1$ be the number of parent smooth blocks out of $c$ number of codes. Let $29 \times c + 60 \times r_1 > 28 \times 4096$ and also let every child block of every smooth parent block be a smooth block. (Note that there are four 4 × 4 child blocks of each $8 \times 8$ parent block.) Then, "single level with classification" scheme with block size 4 × 4 is preferable to "two level with classification" scheme.
- If all the blocks belong to the same class then "classification" is not needed.
- If both smooth and rough type blocks are present (after classification) with number of smooth blocks less than 5% of the total codes present then classification is not needed.

Note that similar inferences may be drawn for the other available fractal-based image coding schemes.

## VI. Conclusions and Discussion

The effectiveness of the GA-based fractal image compression technique depends upon three factors: i) the number of points in the search space $2^l$, ii) the size of the initial population $S$, and iii) the number of iterations $T$. The number of iterations will be different for different images to achieve the near-optimal solution using GA's.

PSNR is used here to measure the quality of the decoded image. There are other measures for judging the quality of the decoded image [18]. It is easy to modify the proposed method for other measures of judging the quality of the decoded image. MSE is used as the fitness function for the set of strings in this article. Any other suitable function, which measures the distortion between the given range block and the obtained range block, can be used as the fitness function instead of MSE in the proposed method. One can achieve a high compression ratio by sacrificing the quality of the decoded image. On the contrary, a high-quality decoded image can be obtained at the cost of compression ratio. Thus, a trade-off has to be made to obtain a good quality decoded image with a considerable amount of compression. A stopping criterion of the decoding process in the present methodology can be suggested by using a threshold value on the difference between the resulting images of two successive iterations.

The GA-based method reported in this article and the quadtree-based method [5] can be applied simultaneously to give rise to a new method of fractal coding. Quadtree can be applied during the partitioning of the given image and GA's can be applied as the search technique in each step of quadtree partitioning.

The threshold value for the classification of range blocks corresponds to the valley in the histogram of the variances of the pixel values of blocks. Note that it can not always be assured that the said histogram is strictly bimodal. The suitable values for the valleys in the histograms are obtained here by visual inspection for the images considered in the present article. One can use any one of the several thresholding methodologies (e.g., [19]) for finding valleys in the histograms.

A simple technique for the classification of the range blocks is used in this article. However, one can use other techniques in this regard. A technique that utilizes the psychovisual properties of human visual system has been developed recently for the classification of the image blocks [20].

The basic philosophy of the proposed GA-based technique can also be adopted for other fractal-based coding techniques. Thomas *et al.* [21] suggested an algorithm for fractal-based image compression in which the neighborhood information plays an important role in increasing the compression ratio. In that method, the domain block for a "seed" range block is found by an exhaustive search mechanism. The domain blocks for the other range blocks are found by utilizing the connectivity of the range blocks. One can adopt the proposed GA-based technique for finding the domain block for the "seed" range block.

## Acknowledgment

The authors are acknowledge Prof. S. K. Pal and S. N. Biswas for their helpful suggestions and encouragement during the course of the work.

## References

[1] M. F. Barnsley, *Fractals Everywhere.* New York: Academic, 1988.
[2] A. E. Jacquin, "Fractal image coding: A review," *Proc. IEEE,* vol. 81, pp. 1451–1465, 1993.
[3] ——, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing,* vol. 1, pp. 18–30, 1992.
[4] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.,* vol. COMM-34, pp. 1105–1115, 1986.
[5] Y. Fisher, E. W. Jacbos, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression,* J. A. Storer, Ed. Boston, MA: Kluwer, 1992, pp. 36–61.
[6] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning.* Reading, MA: Addison-Wesley, 1989.
[7] L. Davis, *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold, 1991.
[8] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs.* New York: Springer-Verlag, 1992.
[9] B. P. Buckles and F. E. Petry, Eds., *Genetic Algorithms.* Los Alamitos, CA: IEEE Comput. Soc. Press, 1992.
[10] S. Forrest, Ed., *Proc. 5th Int. Conf. Genetic Algorithms,* San Mateo, CA, July 1993.
[11] S. K. Pal, D. Bhandari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognit. Lett.,* vol. 15, pp. 261–271, 1994.
[12] S. K. Pal and P. P. Wang, Eds., *Genetic Algorithms for Pattern Recognition.* Boca Raton, FL: CRC, June 1996.
[13] G. A. Edger, *Measure, Topology, and Fractal Geometry.* New York: Springer Verlag, 1990.
[14] J. Feder, *Fractals.* New York: Plenum, 1988.

[15] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 10, pp. 731–747, 1996.
[16] S. Bandyopadhyay, C. A. Murthy, and S. K. Pal, "Pattern classification with genetic algorithms," *Pattern Recognit. Lett.*, vol. 16, pp. 801–808, 1995.
[17] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithm," *Pattern Recognit. Lett.*, vol. 17, pp. 825–832, 1996.
[18] S. Daly, "The visual difference predictor: An algorithm for the assessment of image fidelity," in *SPIE Conf. Human Vision, Visual Processing and Digital Display III*, San Jose, CA, 1992, pp. 2–15.
[19] C. A. Murthy and S. K. Pal, "Histogram thresholding by minimizing gray level fuzzyness," *Inform. Sci.*, vol. 60, pp. 107–135, 1992.
[20] X. Ran and N. Farvardin, "A perceptually motivated three-component image model—Part I: Description of the model," *IEEE Trans. Image Processing*, vol. 4, pp. 401–415, 1995.
[21] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Trans. Image Processing*, vol. 4, pp. 832–838, 1995.

# An Error Resilient Scheme for Image Transmission over Noisy Channels with Memory

Philippe Burlina and Fady Alajaji

***Abstract*— This correspondence addresses the use of a joint source-channel coding strategy for enhancing the error resilience of images transmitted over a binary channel with additive Markov noise. In this scheme, inherent or residual (after source coding) image redundancy is exploited at the receiver via a *maximum a posteriori* (MAP) channel detector. This detector, which is optimal in terms of minimizing the probability of error, also exploits the larger capacity of the channel with memory as opposed to the interleaved (memoryless) channel. We first consider MAP channel decoding of uncompressed two-tone and bit-plane encoded grey-level images. Next, we propose a scheme relying on unequal error protection and MAP detection for transmitting grey-level images compressed using discrete cosine transform (DCT), zonal coding, and quantization. Experimental results demonstrate that for various overall (source and channel) operational rates, significant performance improvements can be achieved over interleaved systems that do not incorporate image redundancy.**

***Index Terms*— Channels with memory, DCT coding, error resilience, joint source/channel coding, MAP decoding, unequal error protection.**

## I. Introduction

We address the problem of the reliable communication of images over bursty channels. Traditional approaches to the design of visual communication systems over noisy channels rely on Shannon's source-channel coding separation principle [9], resulting in what is known as *tandem source-channel coding schemes.* The optimality of this design principle holds only asymptotically; i.e., when no constraints exist on coding/decoding complexity and delay [9]. An alternate approach lies in joint source-channel coding (JSSC): this strategy includes techniques such as *maximum a posteriori* (MAP) detection, channel optimized vector quantization, or adaptive source-channel rate allocation. JSSC has recently received increased attention (e.g., [5], [7], [11]), and has been shown to outperform tandem schemes when delay and complexity are constrained. Most of the work on joint source-channel coding of images [5], [7], [11] has dealt with memoryless channels, disregarding the fact that real-world communication channels—in particular, mobile radio or satellite channels—often have memory.

In this work, we investigate the problem of MAP detection of images transmitted over a binary Markov channel. The MAP detector fully exploits the statistical image characteristics in order to efficiently combat channel noise. It also exploits the larger capacity of the channel with memory as opposed to the interleaved (memoryless) channel. We first describe MAP detection schemes that directly utilize the inherent image redundancy in uncompressed binary images and bit-plane encoded grey-level images. The amount of needed overhead information and the performance degradation when the decoder has imperfect knowledge of the channel parameters are considered.

The MAP detection approach is then validated for systems employing image compression. The residual redundancy of quantized low-frequency discrete cosine transform (DCT) coefficients is exploited via unequal error protection (UEP) and MAP decoding. Experimental results show that the proposed schemes exhibit very good performance, in spite of their low complexity (which primarily resides in the MAP decoder). Specifically, significant gains over systems not exploiting image redundancy can be achieved, at relatively low overall transmission rates.

## II. Channel Model

Consider a binary channel with memory described by $Y_i = X_i \oplus Z_i$, for $i = 1, 2, \cdots$ where $X_i$, $Z_i$ and $Y_i$ represent, respectively, the input, noise and output of the channel. The input and noise sequences are assumed to be independent from each other. The noise process $\{Z_i\}$ is a stationary ergodic Markov process described in [2], with channel bit error rate (BER) denoted by $\epsilon$, where $\epsilon \in [0, 1/2)$, and correlation parameter denoted by $\delta \geq 0$ (the noise correlation coefficient is given by $\frac{\delta}{1+\delta}$). When $\delta = 0$, the channel reduces to the memoryless binary symmetric channel (BSC). The channel transition and marginal probabilities $Q(z_n \mid z_{n-1}) \triangleq \Pr\{Z_n = z_n \mid Z_{n-1} = z_{n-1}\}$ and $Q(z_n) \triangleq \Pr\{Z_n = z_n\}$, are given by

$$\begin{bmatrix} Q(0 \mid 0) & Q(1 \mid 0) \\ Q(0 \mid 1) & Q(1 \mid 1) \end{bmatrix} = \frac{1}{1+\delta} \begin{bmatrix} 1-\epsilon+\delta & \epsilon \\ 1-\epsilon & \epsilon+\delta \end{bmatrix}$$

and $Q(1) = \epsilon = 1 - Q(0)$. Note that this Markov model is *general*; it can represent any irreducible first-order two-state Markov chain. The channel capacity is given [2] by

$$C = 1 - H(Z_2 \mid Z_1) = 1 - (1-\epsilon) h_b\left(\frac{\epsilon}{1+\delta}\right) - \epsilon h_b\left(\frac{1-\epsilon}{1+\delta}\right)$$

where $h_b(\cdot)$ is the binary entropy function. The capacity is monotonically increasing with $\delta$ (for fixed $\epsilon$) and monotonically decreasing with $\epsilon$ (for fixed $\delta$). Note that for fixed $\epsilon$, as $\delta \to \infty$, $C \to 1$.

Manuscript received January 24, 1996; revised May 16, 1997. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Parts of this work were presented at the 1995 International Symposium on Information Theory and the 1996 International Conference on Image Processing. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christine Podilchuk.

P. Burlina is with the Institute for Advanced Computer Studies and the Electrical Engineering Department, University of Maryland, College Park, MD 20742 USA (e-mail: burlina@cfar.umd.edu).

F. Alajaji is with the Department of Mathematics and Statistics and the Department of Electrical and Computer Engineering, Queen's University, Kingston, Ont. K7L 3N6, Canada.

Publisher Item Identifier S 1057-7149(98)02465-8.