IETE Journal of Research
Vol 46, No 5, 2000, pp 279-298

Partitioned Iterative Function System : A New Tool For Digital Imaging

S. K. Mitra, C. A. Murthy and M. K. Kundu

Machine Intelligence Unit
Indian Statistical Institute
203, B. T. Road, Calcutta 700035
Email: {skmitra_t,murthy,malay}@isical.ac.in

This article deals with the fractal based representation of images to perform various image processing tasks like image compression, image magnification and image edge extraction. Representation of an image in terms of a set of affine contractive maps has been regarded as the fractal representation. The set of transformations is called iterative function system (IFS). However, the contribution of the article is based on partitioned iterative function system (PIFS). The basic difference between IFS and PIFS exists in the domain of application of their respective transformations. The cost of generating PIFS code of an image is reduced by using Genetic Algorithms (GAS) based search technique. Image compression is a direct consequence of the PIFS code. On the other hand two other operations *viz.*, image magnification and edge extraction can also be performed using the PIFS coded obtained by the GA based method.

Indexing terms: Iterative function system (IFS), Partitioned iterative function system (PIFS), Genetic Algorithms (GAs), Image compression, Image magnification, Image edge extraction.

1 Introduction

Images are stored in computers in the form of a collection of bits representing pixels (picture elements). The term digital image refers to a two dimensional function defined on a discrete domain and is denoted by $I(x, y)$; the value of I at spatial coordinates (x, y) is known as the pixel value and it gives the light intensity of the image at that point. The representation of images with two dimensional function or a two dimensional array of pixel values is one of the widely used basic forms of representation. Images can also be represented in terms of the parameters of some suitable mathematical models. The most important and direct benefit of representing an image in terms of the parameters of a mathematical model is the reduction in the number of bits required to store images in the computer and thus resulting in reduction of the image data size. This task is known as image compression. The representation of an image in the compressed form should be such that the original form of the image can be reconstructed easily whenever necessary. The process of image compression along with image decompression comprises an important component of digital imaging.

Usually, pixel locations and their values are used as input to a system for performing digital image processing tasks. Thus, if the image is represented in a compressed form, then, an image decompression process should be carried out before taking up any image processing task. The process of decoding requires some computational time leading to the reduction of overall cost and efficiency of the system. An improved image processing system should have the capability of performing image processing tasks using images in the compressed form. Figure 1 shows a block diagram of the existing system and the proposed improved system. Thus, to develop such an image processing system, one should first concentrate upon the representation of images by a suitable mathematical model.

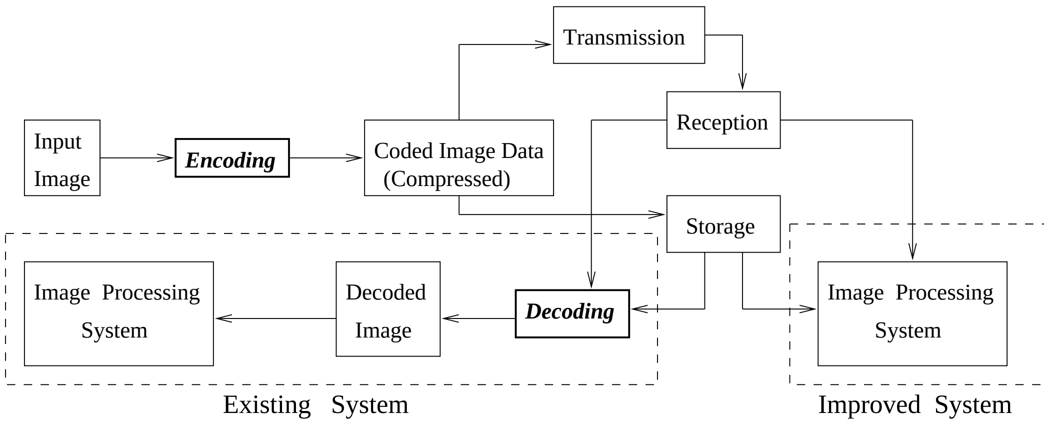


Fig. 1: Block diagram of the existing and the proposed modified image processing system

The present article addresses the task of developing a new improved image processing tool using the theory of fractal for performing image compression, image magnification and edge extraction operations. An image compression scheme using fractal technique and genetic algorithms has been described followed by an image magnification and an image edge extraction scheme. Magnification and edge extraction operations takes the fractal code, which is generated during compression, as input.

The fractal image compression is based on the Iterative function system (IFS). Barnsley, [1] laid the foundation stone of fractal image compression by suggesting Collage theorem [1] of IFS. A fully automated fractal based image compression technique for digital image was first described by Jacquin [2]. In his work, Jacquin partitioned a given image into square blocks called range blocks. The encoding process consists of approximating these blocks from other portions of the image, called domain blocks, through some operations. As a result of this encoding process, separate transformations for each range block are obtained. The set consisting of these transformations, when iterated upon any initial image, will produce a fixed point (attractor) which approximates the target image. This scheme is known as partitioned iterative function system (PIFS). Several algorithms, with different motivations, have been suggested to obtain PIFS or fractal code of a given image. A survey of the image compression methodologies using PIFS has been presented in Section 2. One of the main disadvantage of all these technique is huge computational time requires for encoding. A faster algorithm, to obtain PIFS, using genetic algorithm [3, 4] has been developed, which is faster than the most popular technique proposed by Jacquin [2]. Actually, the efficiency of the proposed algorithm in terms of computational time depends on the selection of genetic parameters.

Genetic algorithms (GAs) [5, 6, 7] are search algorithms which try to emulate biological evolutionary processes to solve optimization problems. Instead of searching one point at a time, GAs use multiple search points. They attempt to find near optimal solutions without going through an exhaustive search mechanism. Thus in effect, GAs have an advantage of achieving a large reduction both in search space and time. Genetic algorithms are used here in finding the appropriate domain block as well as the appropriate transformation for a range block to generate the PIFS code of an image. This article also presents a magnification and an edge extraction algorithm which are embedded in the image reconstruction sequence from the fractal code.

Image magnification, for digital images, is an important task performed with two different motivations . The first one is to increase the image resolution such that the implicit information present in the original can be explicitly visible. The second one is to bring image data, from different sources, to a common scale. The applications of image magnification include satellite image analysis, medical image display and matching of images captured using sensors having different capturing resolutions [8, 9]. Most of the digital magnification techniques use surface interpolation methodologies based on linear, bilinear, cubic or bicubic interpolation [10, 11, 12, 13].

Edge extraction plays a very important role in many image processing applications. Usually, edges in an image are formed due to changes or discontinuities in image intensity. Hence one can say that edge points represent some features of an image. A great deal of effort has been directed towards finding solution to this problem [14, 10], but complete success is yet to be achieved as edge semantics are extremely complicated and is difficult to model. Here we have suggested a completely new approach of edge extraction based on convergence criteria of fractal coders. This is different from conventional approaches that use kernel based gradient computation followed by thresholding [15, 16, 17].

A brief review of the existing methodologies for digital imaging, specially image compression, is described in Section 2. Theory and key features of IFS, image coding and decoding using PIFS are outlined in Section 3. Some basic features of Genetic Algorithms and the methodology of finding fractal code using GAs is described in Section 4. In Section 5 the methodologies for image magnification and edge extraction using fractal code are described. Section 6 presents implementation and results. Discussion and conclusions are provided in Section 7.

2 PIFS Based Methodologies for Digital Imaging

The application of PIFS, so far, is mainly restricted to perform image compression task. To our knowledge, no significant attempt, with a few exceptions, has been made to utilize PIFS (fractal) codes for carrying out image processing tasks other than image compression. This section includes a detailed survey of the literature of PIFS based image compression.

As mentioned earlier, the first implementable PIFS based image compression scheme was proposed by A. Jacquin [18, 19, 2]. Since then, fractal image compression has received considerable attention and most of the fractal compression schemes appeared so far in the literature are based on Jacquin's type compression technique. Most of the schemes of this kind are computationally expensive *i.e.*, the encoding time to generate PIFS codes is usually huge. The problem here is to find a suitable approximation of an image by a set of affine contractive maps [20]. A good collection of both theoretical and application aspects of several fractal image compression schemes is available in [21, 22].

The scheme due to Jacquin consists of finding suitable transformations defined on image blocks. These transformations are affine in nature, as approximation of smaller image blocks (range blocks) are carried out by applying transformations on suitably chosen larger image blocks (domain blocks). As the selection of appropriate domain blocks and transformations usually takes a huge computational time, he has suggested a trimming of search space following a classification scheme well known in image processing [23]. Larger image blocks are stored into a set of categories namely shade blocks, edge blocks and midrange blocks. Same classification has been employed to classify the range blocks too. For each range block, domain blocks of the same category are searched, and that domain block which suitably approximates the candidate range block under an affine contractive map, is found. The preciseness of the approximation is judged by root mean squared error (RMSE). Note that the information regarding the classification scheme has to be included in fractal coders so as to regenerate the decoded image.

The literature of PIFS based image compression has addressed mainly the problem of suggesting an effective algorithm for finding a set of suitable transformations, called fractal coder, which can approximate the image. In regards to this, several methodologies have been suggested with mainly three motivations namely (1) improving the performance of the fractal coder, (2) reduction in the search space for finding the suitable set of transformations, and (3) analyzing the convergence properties of the fractal coder.

Many of the techniques of fractal image compression deal with either improvement in the compression ratio [24] or improvement in the quality of the decoded image [25]. The number of transformations required to approximate the given image depends on the image complexity. For a given image, large number of transformations lead to better fidelity but poor compression ratio. To make a trade-off between compression and fidelity, Fisher et al. [26] have suggested two different approaches to encode a given image. Target of the first approach is to achieve more compression keeping the number of range blocks fixed up to a prefixed level in a quadtree partitioning or in a horizontal vertical (HV) partitioning. On the other hand, the second approach consists of partitioning the image, either in quadtree or in HV, up to that level at which the desired fidelity is achieved. In quadtree partitioning, every square partition may be subdivided into four smaller squares with one fourth the area of original square. The partitioning of images in quadtree approach can also be performed using the variance information of subimages [27]. A recursive partitioning of the image along horizontal or vertical lines is called HV partitioning. A mixture of triangular and rectangular partitioning of images has also been tried for the same purpose [28]. The other partitioning scheme used in fractal image compression is lapped partitioning [29]. It is to be noted that, besides the information regarding location of selected domain blocks and the parameters of the selected maps, the information regarding partitioning scheme is also required to be stored. This is true for all the schemes of this kind to reproduce the image whenever necessary. On the other hand, in non overlapping square partitioning, no extra bit is required to store the information regarding partitioning scheme. Thus a complicated partitioning scheme may reduce the compression ratio compared to that of square partitioning.

Fisher et al. have also pointed out another important phenomenon regarding the contractivity factor of the set of maps and coined the term eventual contractivity [26]. It has been observed, from the experimental results, that it is not necessary to impose strict contractivity on the transformations since the contractivity of their union is a sufficient condition to ensure the convergence of the iteration procedure during decoding [26]. A generalized Collage theorem of an IFS, in the context of eventual contractive maps, has been suggested by Fisher et al. [21]. But the convergence procedure of PIFS scheme, theoretically, in the context of eventual contractivity has not been studied.

Some block based fractal image compression techniques utilize irregular shaped blocks. In this direction, a region based partitioning scheme of the image has been suggested by Thomas and Deravi [30]. The algorithm starts with defining a seed range block among all the range blocks, and searching for appropriate domain to range transformation. The algorithm then attempts to extend the seed range block in all four principle directions. In the extended form, the parameters of the extended transformation are either same as that of seed or may be selected on an adaptive basis. The selection of fractal codes using an adaptive technique is also available in [31].

A different scheme for encoding images by PIFS using variable block sizes has been proposed by Tanimoto et al. [32]. In this scheme the range blocks are determined by a split and merge method. In this method, range blocks of various sizes and various shapes are extracted from the given image. The concept of variable size blocks has been extended by Ruhl et al., results in an adaptive partitioning of images to determine the range blocks. In this algorithm, the starting range blocks are small square blocks. Now, for the merging of the starting range blocks, an evolutionary programming [33] and its deterministic version [34] have been adopted. Another significant contribution, along this line, is the overlapped adaptive partitioning of image blocks, suggested by Reusens [35]. This overlapped partitioning is embedded in the quadtree segmentation of the image support. The region of overlap is a function of the block size at each level of the quadtree segmentation. During decoding, the value of a pixel corresponding to an overlapped region is computed as the weighted sum of the different contributions leading to that pixel.

To solve the issue of improving the quality of the decoded image, visual perception has been used in the context of searching the matched domain blocks and the transformations [36]. An entropy based constraint has also been used in this regard [37]. Unlike exploiting the similarity between range blocks and domain blocks, Rinaldo and Calvagno [38] designed an image coder in which the similarity of different subbands in a multiresolution decomposition of the image is exploited. In a quadtree partitioning approach, the proposed coding scheme consists of approximating blocks in one subimage, from blocks in another subimage of the lower resolution with the same orientation. The transformations defined for this purpose are similar to those used in classical fractal block coders but need not be contractive. A generalized version of the fractal technique for image compression has been discussed in [39].

Significant contributions, in the literature of fractal image compression using PIFS, have also been made by suggesting computationally efficient algorithms. The problem of expensive computational cost has been taken care by adopting various search mechanisms for finding appropriate domain blocks and the appropriate transformations. Another solution towards the same is to restrict the selection of a domain block, as a candidate for matching, by adopting either suitable classification or clustering techniques or imposing geometric constraints. A nearest neighbour search procedure has been adopted by D. Saupe [40]. In particular, a relation between fractal image compression technique and the multidimensional nearest neighbour search has been found.

A different kind of search mechanism, known as pyramid search, has been used in fractal image compression [41, 42]. The algorithm proposed by Lin and Venetsanopoulos [41] uses a pyramid search which is embedded in a quadtree partitioning. Assuming the matching error, *i.e.*, error in approximating the range blocks, to be an independent, identically distributed (i.i.d.) Laplacian random process, the threshold sequence for the objective function to minimize the error, in each pyramidal level, is derived. The computational efficiency depends on the depth of the pyramid.

It should be noted that while suggesting a search algorithm for selecting appropriate transformations as well as appropriate domain blocks for range blocks, utmost care should be taken to reduce the complexity of the algorithm as much as possible. Moreover, a complex search procedure may be computationally expensive. In such a case, the whole encoding process may be a computation intensive procedure.

The complexity reduction of the search procedure of fractal image compression using clustering of the search space is an important issue and attracts considerable attention. Lepsoy and Oien [43] have suggested an iterative algorithm to cluster the codebook, in an adaptive way. The procedure consists of subdividing the codebook into clusters with the help of cluster centers. In the encoding process, the cluster center that most resembles the range block under consideration is found and it determines the cluster to be considered. Within this cluster, the codebook block that most resembles the range block is to be found. Clustering of domain blocks using data structures, *e.g.*, k-d trees [44], r-trees [45], continuous features [46], geometric constraints [47, 48], are also available in the literature. Some other techniques of this kind can be found in [49, 50, 51, 52, 53].

Note that, to use a clustering algorithm to subdivide the codebook blocks, proper number of clusters has to be determined. Moreover, a proper function to measure the dissimilarity between blocks is to be defined. The clustering procedure also needs to be validated [54]. Without the fulfillment of above points, the results of a clustering algorithm may not be useful for the purpose of representing a search space and hence reducing it.

Comparatively less attention has been paid towards the problem of analyzing the convergence phenomenon of the fractal coders. The algorithms for fractal image compression using PIFS have been suggested on the basis of an inherent assumption that the convergence procedure of PIFS is same as that of IFS. But, in reality, there are differences between IFS and PIFS mostly in the application domain. Thus, a study on the convergence procedure of the PIFS based fractal codes in the context of image compression needs to be carried out. Two different approaches for studying the convergence of affine fractal operators have been suggested by W. Skarbek. In the first approach [55], effective sufficient conditions for eventual contractivity have been presented and the notion of perceptual convergence has been introduced in the case of non contractive fractal transforms. Moreover, convergence, with fewer number of iterations, of the quantized version and the mean shifted version of the fractal transform have been studied. In the second approach [56], the convergence has been investigated by an analysis of block influence graph and pixel influence graph. The graph stability condition in pixel influence graph appears to be sufficient and necessary for convergence of selecting fractal transforms. Another graph theoretical analysis of fractal coders has been presented in [57]. Other significant contributions towards this problem have been made by Hurtgen. The convergence of fractal coder concerning a necessary and sufficient condition based on the spectral radius of the transformation matrix has been investigated by Hurtgen and Simon [58]. The convergence of the fractal transform defined on the finite dimensional vector space for signal modelling has been reported by Hurtgen and Hain [59]. A significant contribution, considering the wavelet based framework, for analyzing block based fractal compression schemes has been made by G. Davis [60, 61, 62]. In particular, it has been shown that the fractal block coder of the form of PIFS is analogous to a wavelet subtree quantization scheme. The effectiveness of the performance of fractal block coders has been investigated in a wavelet framework. Some other works on the convergence aspects of fractal image compression are available in [63, 64, 65, 66, 67].

There are some other methodologies which utilize fractal with other compression methodologies result in new image compression techniques. Actually this kind of encoding mechanism is better known as hybrid coding. In hybrid coding more than one approach is used in a single algorithm to result in better compression or better fidelity. A large number of contributions which utilize a fractal technique and other techniques such as vector quantization [68, 69], neural networks based coding [70, 71], wavelet based coding [72], DCT based coding [73, 74], transform coding [75] and others [76, 77, 78, 79], are available in the literature. Comparative studies between fractal image compression method and other image compression methods has been presented in [80, 81, 82, 83].

There are other fractal based image compression methods which do not use the approach based on PIFS. Most significant approaches among these are given in [84, 85, 86, 87]. The fractal image coding algorithm proposed by Vines [87] is based on an orthonormal basis approach which is a hybrid method combining principles of transform coding with those of fractal encoding. The method suggested by Dudbridge [84] begins with a nonoverlapping square partition. An interesting investigation was carried out by Lin and Venetsanopoulos [85] by incorporating nonlinear contractive functions. Introduction of the basis orthogonalization in the fractal encoder and decoder process is the novel idea suggested by Oien and Lepsoy [86]. The benefits of such a scheme include minimum number of iterations to get the attractor of the fractal coder. Another scheme with non iterative decoder was suggested by Kim et al. [88]. A scheme with different decoding procedure has also been suggested by Hamzaoui [89].

So far, we have discussed various methodologies of image compression based on PIFS. It has been mentioned that there are only a few methodologies available for carrying out tasks like image magnification edge extraction and segmentation. In fact, this area is much neglected. Significant results are yet to be obtained.

3 Mathematical Foundation of Fractal Image compression

The detailed mathematical description of the IFS theory, Collage theorem and other relevant results are available in [1, 90]. Only the salient features are discussed here.

3.1 Iterative Function System (IFS)

Let I be a given image which belongs to the set X. Generally X is taken as the collection of compact sets. Hence I can be looked upon as a set belonging to X. Our intention is to find a set $\mathcal{F}$ of affine contractive maps such that its fixed point will be a close approximation of the given image I. The fixed point or attractor "A" of the set of maps $\mathcal{F}$ is defined as follows :

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

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

$$\begin{aligned} \mathcal{F}^N(J) &= \mathcal{F}(\ \mathcal{F}^{N-1}(J)\ ), \text{with} \\ \mathcal{F}^1(J) &= \mathcal{F}(J), \quad \forall J \in X. \end{aligned}$$

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

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

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

$$d(I, \mathcal{F}(I)) \leq \epsilon \tag{2}$$

where ϵ is a small positive quantity. Now, by Collage theorem [1], it can be shown that

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

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

From equation (3) it is clear that, after a sufficiently large number of iterations, the set of affine contractive maps $\mathcal{F}$ produces a set which 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 code for the given image I.

3.2 Image Coding Using PIFS

The structure of PIFS codes are almost same as that of IFS codes. The only difference is that PIFS codes are obtained and applied to a particular portion of the image instead of the whole image.

Let I be a given digital image having size $w \times w$ and the set of gray levels be $\{0, 1, 2, \cdots, l-1\}$. Thus the given image I can be expressed as a matrix $((g(i,j)))_{w \times w}$, where i and j stand for row number and column number respectively and $g(i,j)$ represents the gray level value for the position (i,j). The image is partitioned into n non overlapping squares of size, say $b \times b$, and let this partition be represented by $\mathcal{N} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is named as range block. Note that the number of range blocks is $n = \frac{w}{b} \times \frac{w}{b}$. Let $\mathcal{M}$ be the collection of all possible blocks of size $2b \times 2b$ within the image. Let $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Here $m = (w - 2b + 1) \times (w - 2b + 1)$ and $\mathcal{D}_j$'s are named as "domain blocks".

Now, let us define,

$$\mathcal{A} \;=\; \{1, 2, \cdots, w\} \times \{1, 2, \cdots, w\} \times \{0, 1, 2, \cdots, l-1\}.$$

Here $\mathcal{A} \subset \mathrm{I\!R}^3$. Note that any image I is a subset of $\mathcal{A}$ but any subset of $\mathcal{A}$ is not necessarily an image. Also $\mathcal{R}_i \subset \mathcal{A}; \forall i$ and $\mathcal{D}_j \subset \mathcal{A}; \forall j$.

Let, for a range block $\mathcal{R}_i$,

$$\mathcal{F}_j = \{f : \mathcal{D}_j \rightarrow \mathcal{A} \ \ ; \ \ f \ \text{ is an affine contractive map}\}.$$

Let, $f_{i|j} \in \mathcal{F}_j$ be such that

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

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

Here "d" is a suitably chosen distance measure. The distance measure "d" used here is taken to be the simple Root Mean Square Error (RMSE) between the original set of gray values and the obtained set of gray values of the concerned range block. The map from $\mathcal{D}_k$ to $\mathcal{R}_i$ is constructed in such a way that the pixel positions of $\mathcal{R}_i$ and $\widehat{\mathcal{R}}_{i|k}$ are the same. The difference between $\mathcal{R}_i$ and $\widehat{\mathcal{R}}_{i|k}$ are found only in the gray level values of the pixels. RMSE has been used here for measuring the distortion and it serves the purpose of a distance measure. It has been seen that, for the proposed encoding scheme, attractor - which is close to the original image - may be obtained with the help of RMSE [91].

Now let k be such that

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

The aim here is to find $f_{i|k}(\mathcal{D}_k)$ for each $i \in \{1, 2, \cdots, n\}$. In other words, for every range block $\mathcal{R}_i$, one needs to find an appropriately matched domain block $\mathcal{D}_k$ as well as an appropriate transformation $f_{i|k}$. The set of maps $\mathcal{F}$ thus obtained is called the partitioned or local IFS or fractal code of image I. Figure 2 illustrates the mapping of domain blocks to the range blocks.

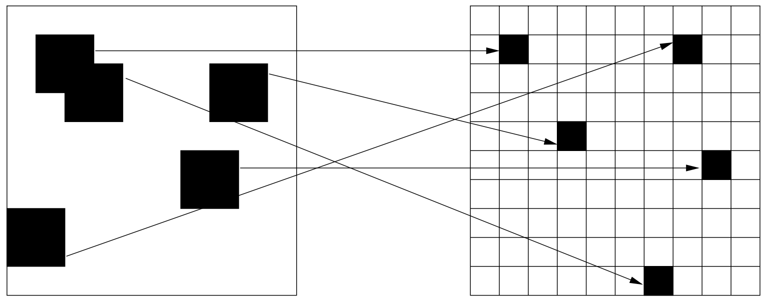


Fig. 2: Mapping from Domain block to Range block

To find the best matched domain block as well as the best matched transformation, all possible domain blocks as well as all possible transformations are to be searched with the help of equation (4). The problem of searching for an appropriately matched domain block and transformation for a range block can be solved by enumerative search [2] and by using Genetic Algorithms [4]. We have considered here eight isometric transformations as described by Jacquin [2] to search out the appropriate transformation for a range block. The corresponding parameters of the transformation are selected by fitting a straight line between two sets of gray values, one from range block and the other is from the matched domain block.

3.3 Image decoding using PIFS

Let us consider the generated PIFS or fractal code be $\mathcal{F}$ of a given image I. The natural decoding scheme simply consists in iterating the code $\mathcal{F}$ on any initial image, say I_0, until convergence to a stable decoded image is observed. Thus, the decoding process is of the form $\{\mathcal{F}^N(I_0)\}_{N>0}$. At the Nth iteration, the image I_N is used as input to the decoding system, where I_N is the output image obtained from the $(N-1)th$ iteration. In particular for each index i, the transformation $f_{i|k}$ is applied to the domain block $\mathcal{D}_k$ of the current input image, and mapped on to the range block $\mathcal{R}_i$ of the current output image which is input image of the next iteration.

In the next section the basic principles of Genetic Algorithms (GAs) and the methodology for finding fractal codes using GAs are described.

4 Genetic Algorithms (GAs)

First GAs are described in brief, so that it will be convenient to understand the methodology for finding fractal code using GAs.

GAs are adaptive search processes based on the notion of selection mechanism of natural genetic system [5]. To solve an optimization problem, a 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 a chromosome. Usually, the chromosomes are strings of 0's and 1's. If the length of string (chromosome) is L then the total number of possible strings is 2^L. 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 [5].

Three basic genetic operators, i)Selection, ii)Crossover and iii)Mutation are exploited in GAs. Details description of these operations are available in [5, 6]. The strings in an initial pool are judged by their respective fitness values and marked accordingly in selection operation. Next, after forming a new population using selection process, crossover and mutation operators are used one by one to generate a new population of the same size (S). Three operators are then 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 termed as iteration. We have used here the elitist model [7] of GAs which keeps track of the best string obtained so far. In the present work, the number of iterations, say T, is fixed apriori for the termination of GAs. Note that It has been shown that, as the number of iterations in a GA goes to infinity, one can always find the optimal solution [92]. Various applications of GAs in digital imaging are also available in [93, 94, 95].

Now let us describe the methodology for finding fractal code using the principle of GAs.

4.1 GAs to find PIFS code

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

The number of possible domain blocks to be searched are $(w-2b+1)\times(w-2b+1)$ (Section 3.2). The number of transformations to be searched for each domain block is 8 (Section 3.2). Thus the space to be searched consists of M elements. M is called the cardinality of the search space. Here $M = 8\ (w-2b+1)^2$. Let the space, to be searched, be represented by $\mathcal{P}$ where

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

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 of L depends on the values of w and b. The fitness value of a string is defined to be the RMSE 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. Initially, S strings are selected randomly from 2^L strings, to form an initial population for GA. The various steps of the GA, as mentioned above are implemented repeatedly up to T iterations. Note that the total number of strings searched up to T iterations is $S \times T$. Hence, $\frac{M}{S\ T}$ provides the search space reduction ratio for each range block.

The set of range blocks are grouped into two sets according to the variability of the pixel values in these blocks [4]. 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 threshold, we call the block as smooth type range block. Otherwise we call it a rough type range block. 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.

Now, once the PIFS code is obtained, rest is to use it as an input to the improved image processing system. In this system, either one can simply decode the image or can perform image magnification or edge extraction which ever is needed.

In the following Section, methodologies for image magnification and edge extraction are described.

5 Other Applications of Fractal Techniques

5.1 Image Magnification

Let, as before, I be a given image having size $w \times w$ and the range of gray level values be $\{0, 1, 2, \cdots, l-1\}$. Let the set of transformations $\mathcal{F}$ is the obtained PIFS code of I. It has already been pointed out in Section 3.2 that there exists the attractor of this PIFS code.

Thus we have,

$$d(\bigcup_{i=1}^{n} \mathcal{R}_i, \ \bigcup_{i=1}^{n} \widehat{\mathcal{R}}_{i|k} \ ) \ = \ d(\bigcup_{i=1}^{n} \mathcal{R}_i, \ \bigcup_{i=1}^{n} f_{i|k}(\mathcal{D}_k \ ) \ \leq \ \epsilon_1, \tag{5}$$

where ϵ_1 is a small positive quantity.

The transformation $f_{i|j}$ is such that $f_{i|j}(\mathcal{D}_j)$ approximates $\mathcal{R}_i$. $f_{i|j}$ consists of two parts, one for spatial information and the other for information of gray values. The first part indicates which pixel of the range block corresponds to which pixel of domain block. This part can be achieved by using any one of the eight transformations on the domain blocks [2, 4]. The second part is to find the scaling and shift parameters for the set of pixel values of the domain blocks to the range blocks. This part is obtained using least square analysis of two sets of gray values once the first part is fixed. To perform the least square, the domain block is contracted [91] such that its size becomes equal to the size of the range block.

Thus $f_{i|k}$ can be looked upon as mixture of two transformations, $f_{i|k} \ = \ t_{i|k} \circ \mathcal{C}$, where, $\mathcal{C}$ is contraction operation and $t_{i|k}$ is transformation for rows, columns and gray values.

Now to perform magnification task using the given image I, one needs to define a magnification operator, say, $\mathcal{O}$ such that

$$\rho(\bigcup_{i=1}^{n} \mathcal{R}_i, \ \bigcup_{i=1}^{n} \mathcal{O}(\mathcal{R}_i) \ ) \ \leq \ \epsilon_2 \tag{6}$$

where ϵ_2 is a small positive quantity.

As the process of magnification ends up with a magnified image, the distance measure ("ρ"), used in this set up, should be selected in such a way that it can compute the distortion between two images with unequal sizes. Note that, though RMSE can be chosen as a suitable distance measure for image compression, the same measure is not suitable for the present magnification task. RMSE can measure the distortion between two images having equal sizes. Thus a proper selection of "ρ" is needed for the present case to measure the distortion between two images having unequal sizes.

Now, from (6) and using the distance measure "ρ", we have,

$$\rho(\ \mathcal{R}_i, \ \widehat{\mathcal{R}}_{i|k} \ ) \ \leq \ \epsilon_{i1} \tag{7}$$

Where ϵ_{i1} is a small positive quantity. Now by (5), (6) and (7) we have,

$$\rho(\bigcup_{i=1}^{n} \mathcal{R}_i, \ \bigcup_{i=1}^{n} \mathcal{O}(\widehat{\mathcal{R}}_{i|k}) \ ) \ \leq \ \epsilon_3 \tag{8}$$

where ϵ_3 is a small positive quantity. Again, we have,

$$\widehat{\mathcal{R}}_{i|k} \;=\; f_{i|k}(\mathcal{D}_k) \;=\; (t_{i|k} \circ \mathcal{C})(\mathcal{D}_k).$$

So,

$$\rho(\bigcup_{i=1}^{n} \mathcal{R}_i, \bigcup_{i=1}^{n} (\mathcal{O} \circ t_{i|k} \circ \mathcal{C})\ (\mathcal{D}_k)\ ) \quad \leq \quad \epsilon_3. \tag{9}$$

Now, reconstruction of images using the magnification operator $\mathcal{O}$ should be an inverse of the contraction operation using the operator $\mathcal{C}$. So, by (9)

$$\rho\ (\ \mathcal{O}\ (\widehat{\mathcal{R}}_{i|k})\ ,\ t_{i|k}(\mathcal{D}_k)\ ) \leq \epsilon_4. \tag{10}$$

Hence, by (8),

$$\rho\ (\ \bigcup_{i=1}^{n} \mathcal{R}_i\ ,\ \bigcup_{i=1}^{n} t_{i|k}(\mathcal{D}_k)\ ) \leq \epsilon_5. \tag{11}$$

Both ϵ_4 and ϵ_5 are small positive quantities.

From equation (11), it is clear that there is no need of constructing the magnification operator $\mathcal{O}$, only the second part of the fractal codes has to be applied on the domain block to get an image which is very close to the given image I and this image has size $2w \times 2w$.

The most important factor involved in a magnification task is the order of magnification. In the present case the order of magnification is defined as follows. Let an image I having size $w \times w$ is used as an input to a magnification system. Now if the output image $\hat{I}$ (say) is of size $kw \times kw$, then k is called the order of magnification. Usually k is taken to be an integer. Here, the term magnification factor is also used to indicate the order of magnification. So far we have discussed how to apply the fractal codes to get a magnified image which is magnified by a factor 2. On successive applications of this proposed algorithm, magnification by factor 4, 8, 16 etc. can also be achieved. Here the term successive magnifications implies dividing the task of magnification by a factor, which is a power of 2, into several steps. In particular magnification of an image by a factor 2^n (n is a positive integer) is divided into n steps. In the first step, magnified image is obtained from PIFS code considering $n = 1$. Then in the next step, *i.e.*, in the case of $n = 2$, magnification task is performed using the PIFS code and the information regarding the already obtained magnified image in the previous step. Incrementing the value of n at each step and proceeding in a similar way, the required magnified image is obtained. The details of successive magnification is available in [96, 97].

Once the magnified image is obtained, the performance of the magnifier is judged by the distortion measure based on the edge pattern of the images as described in [96].

In the next subsection edge extraction from the reconstruction sequence of fractal codes is presented.

5.2 Image Edge Extraction

Edge extraction using PIFS code is a sequential process. In this sequential process, edges are extracted from the image decoding process (described in Section 3.3). In particular, a decoding process is carried out with some modifications. Now for the better understanding of the process and to know the motivation behind such an algorithm one should know the convergence procedure of PIFS to a stable image through an iterative sequence. The detailed description of the mathematical formulation of the convergence of PIFS is given in [91, 98]. We have furnished here only the relevant portions for convenience.

5.2.1 Convergence of PIFS

Let I be a given image having size $w \times w$, and the set of gray level values be $\{0, 1, 2, \cdots, l-1\}$. For this given image, we can construct a vector $\underline{x}$ whose elements are the pixel values of the given image I. Note that there are w^2 pixels of I. Thus,

$$\underline{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots \quad , x_{w^2})'$$

is the given image where x_1 is the pixel value corresponding to the $(1,1)th$ position of I. Likewise, let x_r be the pixel value corresponding to the $(i,j)th$ position of I, where,

$$r = (i - 1)\, w + j, \quad 1 \leq i, \; j \leq w.$$

In this set up PIFS can be viewed as follows. There exists an affine (linear), not necessarily strictly contractive, map for each element of $\underline{x}$ and this map is called forward map of the element. In the process of iteration, the input to a forward map will be any one of the w^2 elements of $\underline{x}$ and the map is called backward map for this input element. Thus for each element of $\underline{x}$ there exists a forward map and an element of $\underline{x}$ can have one or more or no backward map. The set $\mathcal{F}$, of forward maps, is called the PIFS codes of I.

Let f_1 be the forward map for a particular element x_{r_1}, where $r_1 = (i_1 - 1)w + j_1$. Also let this element be mapped from the element x_{r_2}, where $r_2 = (i_2 - 1)w + j_2$. Thus f_1 is the backward map for x_{r_2}. Again x_{r_2} is being mapped from x_{r_3}, $(r_3 = (i_3 - 1)w + j_3)$ with a forward map f_2. Thus we have a sequence of maps for the element x_{r_1} as following.

$$(i_1, j_1) \xleftarrow{f_1} (i_2, j_2) \xleftarrow{f_2} (i_3, j_3) \xleftarrow{f_3} \cdots \xleftarrow{f_{m-1}} (i_m, j_m); \; m \leq (w^2 - 1). \tag{12}$$

The above sequence will be stopped at $(\, i_m, \; j_m \,)$ if

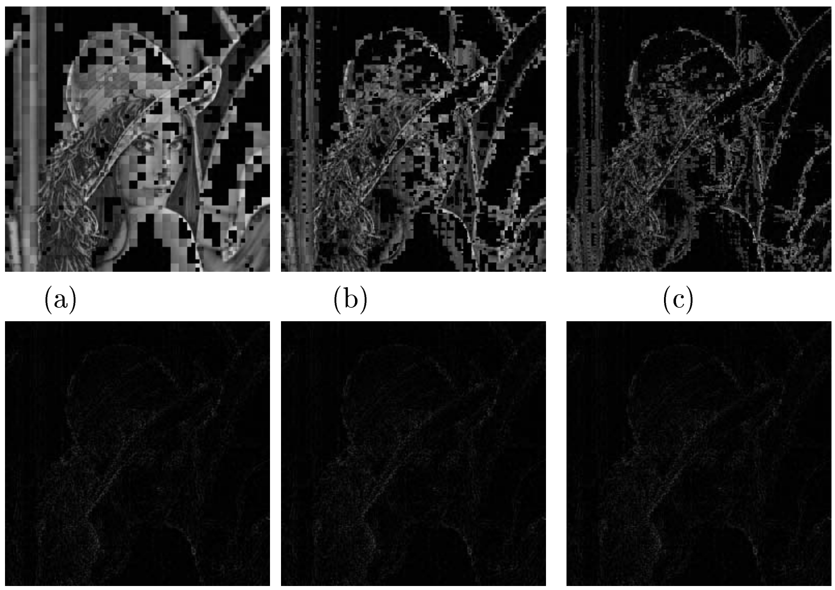
$$(\, i_{m+1}, \; j_{m+1} \,) = (\, i_k, \; j_k \,); \quad \text{for } k = 0 \text{ or } 1 \text{ or } 2 \text{ or } \cdots \text{ or } m. \tag{13}$$

The stopping phenomenon of this sequence is mandatory as there are finite number (w^2) of elements in $\underline{x}$. It is clear from the sequence (12) that during the iterative process the element x_{r_1} will have a fixed point once the element x_{r_2} is fixed. Again the convergence (to a fixed point) of the element x_{r_3} confirms the convergence of the element x_{r_2} and likewise for the rest of the elements. Thus convergence of the last element of the sequence implies the convergence of the rest of the elements. The convergence of the last element of the sequence is possible in four different ways according to the stopping condition (13). The detailed description of these four cases and their convergence which is not needed for the present problem, can be found in[91]. The most important thing to be noted is that, for each element, there will be a sequence of the form (12) and each element has fixed points.

5.2.2 Construction of Edge Image using PIFS

So far, we were able to establish that, by construction, PIFS code is such that each element of the given image has a sequence of the form (12). Let us consider the number of elements included in a sequence as the length of the sequence. So, from (12) it is quite evident that different elements will have different sequences with different lengths. Now since each of the elements has a fixed point [91], the time taken to converge will be different for different elements.

Note that the elements (pixels) which are included in the smooth type range block will converge right after first iteration. On the other hand, the edges, which are of main interest here, are included in the rough type range blocks. In the process of encoding, rough type range blocks will match with domain blocks containing edges. Moreover matching of an edge pixel is restricted to only edge pixels as we are interested in finding the self similarities present in the image. In a rough type range block, in general, there will be both edge pixels and non edge pixels. We want to approximate all the pixels through a contractive continuous map. The closeness of the estimator with a pixel value will depend not only on the map but also on the complexity of the pixel (i.e., the pixel value in relation to its surrounding pixel values) from where it is being mapped. As one edge pixel is mapped to another edge pixel, one map is not sufficient to carry the information regarding an edge pixel. An edge pixel, thus, is approximated by a sequence of maps. Hence, edge pixels will require more iterations to converge compared to non edge pixels. Hence, on the basis of convergence time, we can classify the pixels as edge and non edge.



(a) (b) (c)

(d) (e) (f)

Fig. 3: Difference images of "Lena" after 1st (a), 2nd (b), 3rd (c), 8th (d), 9th (e) and 10th (f) iterations with the "Rose" image (Figure 6) being the starting image

In support of our claim, we illustrate another phenomenon which occurs during the process of decoding. The decoding process of "Lena" image from its PIFS code has been stopped after ten iterations as no visual difference has been found in successive iterations [4]. But from the difference images, as shown in Fig. 3, it is observed that strong edges are prevalent, though with low intensity, even after ten iterations. These difference images are obtained by taking difference between original image and images obtained during decoding process. This phenomenon experimentally supports our claim that edge pixels take more time to converge than non edge pixels.

To locate edge pixels we have attached penalties to pixels, depending on their convergence status during decoding. The more the number of iteration to converge, the more is the penalty attached to that pixel. Note that, PIFS code is nothing but a set of contractive maps which are continuous. But, edges present in the image are discontinuities due to sudden changes in gray values. During the process of decoding, PIFS code needs to grab those discontinuities by continuous maps. The process of approximating discontinuities by continuous maps is necessarily slow, *i.e.*, it takes more iterations to give rise to a better approximation. Hence, pixels having large penalty values would be edges pixels.

A very simple penalty function scheme using the number of iterations, which depends on the status of the convergence has been designed. Once the pixel value converges to a fixed point, no further penalty is imposed. On the other hand penalty of each iteration is added with the penalty that already exists for a pixel if the pixel value has not converged yet. The penalty values for the first few iterations need to be small to let the non edge pixels to converge. Similarly, the penalty values for the last few iterations need to be small so that the impact of the strong edge pixels on the other edge pixels would be less. These penalties for the pixels, once the number of iterations for reconstruction is fixed, would indicate the degree to which a pixel may be designated to be an edge pixel. Thus, the edge image is nothing but an image where the gray value of a pixel is nothing but its penalty value. So we will fix two iteration values which will be considered as lower and upper bounds and within these bounds, penalty values will be high. To represent this phenomenon, an S type penalty function scheme is used. The mathematical formulation of the penalty function is as follows.

Let $p(i,t)$ be the penalty function for a pixel at iteration i. Here t is difference of pixel values between ith and $(i-i)th$ iteration. $p(i,t)$ is defined as follows.

$$\begin{aligned} p(i,t) &= \quad 0 & ; \quad i = 0\ ,\ \ \forall t \\ p(i,t) &= \quad p(i-1,t) & ; \quad i \geq 1\ ,\ \ t < \delta \\ &= \quad p(i-1,t) + \sum_{j=1}^{i} j * c(j) & ; \quad 1 \leq i \leq N\ ,\ \ t \geq \delta \end{aligned}$$

where the coefficient $c(j) \in [0,1]$ is computed from a Π type function [99] which is as follows

$$\begin{aligned} c(j) &= \quad 1 - 2 * d(j)^2 & ; \quad 0 \leq d(j) < \frac{1}{2} \\ &= \quad 2\ (1 - d(j))^2 & ; \quad \frac{1}{2} \leq d(j) < 1 \\ &= \quad 0 & ; \quad Otherwise \end{aligned}$$

where

$$d(j) = \frac{|j - \frac{L_1+L_2}{2}|}{L_2 - L_1}.$$

Here,

$$\begin{aligned} \delta &= \text{Threshold value of } t \text{ beyond which penalty is non zero} \\ L_1 &= \text{Lower bound for the number of iterations} \\ L_2 &= \text{Upper bound for the number of iterations} \\ N &= \text{Number of iterations at which sequential process is stopped} \end{aligned}$$

These parameters δ, L_1 and L_2 act as control parameters of the proposed scheme. Incorporating the penalty function in the reconstruction process, an edge image of the original image can be obtained.

Now, it is observed that, in many cases, the output edge image which is obtained from reconstruction sequence with penalty function may have discontinuity in edges. A simple edge linking algorithm as described in [96] is used to get the final edge image.

The results of the specific implementation of the aforesaid algorithms of image compression, magnification and edge extraction are given in the next Section.

6 Implementation and Results

The results of the proposed methodologies are given below. Algorithms have been tested on several images. But the results of the "Lena" image have only been shown here. "Lena" is the most widely used image in the image processing community.

6.1 Image Compression

The GA based method discussed in Section 3.2 and 4.1 is implemented on 256×256, 8 bit/pixel "Lena" image. To make the encoding process faster, the image is subdivided into four 128×128 subimages, each of which is encoded separately [2, 4]. Also a two level partition [2, 4] scheme is adopted for the specific implementation of the GA based Scheme.

First of all, to classify range blocks (Section 4.1), 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.

Considering parent range blocks of size 8×8 and child range blocks of size 4×4 and using two level image partition scheme [2], each subimage is then encoded. GAs 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 $(128-16+1) \times (128-16+1) = 113 \times 113$ and $(128-8+1) \times (128-8+1) = 121 \times 121$ for parent and child range blocks respectively. Thus the cardinalities (M) of the search spaces for these two cases are $113 \times 113 \times 8$ and $121 \times 121 \times 8$ 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 would replace it. Out of these 2^{17} binary strings, 6 strings ($S = 6$) are selected randomly to construct an initial population. A high probability is taken for the crossover operation. For mutation operation, P_{mut} (mutation probability) is varied over the iterations. 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.

The algorithm is also tested on several other images. These images are also 8 bit/pixel image and of size 256×256. In this cases, the starting image is the "Rose" image as shown in Figure 6. The compression ratios and PSNR values of the test results are shown in Table 1. The search space reduction ratios for these images are same as that of the "Lena" image since the sizes of parent and child range blocks and the values of S and T are the same.

Table 1: Test results of the GA based method for 256×256, 8 bit/pixel images

Image	Compression Ratio	Bits/ pixel	PSNR (in db)
Lena	10.50	0.76	30.22
Girl	11.37	0.70	30.74
Seagull	7.40	1.08	27.27
LFA	5.51	1.45	26.86

The diagrams for the original and decoded "Lena" images are shown in figures 4 and 5 .

Figure 6 shows an arbitrary starting image of "Rose" for the reconstruction of the fractal code of "Lena". The reconstruction sequence is stopped after ten iterations.

Fig. 4: Original Lena image (8 bpp)

Fig. 5: Decoded Lena image (0.76 bpp)

The GA based technique of fractal image compression method as compared with exhaustive search mechanism, is found to be faster. The search space is reduced at the order of 20, in case of GA based 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 for GAs to perform well for optimization problems [100].

The obtained results for "Lena" image are also comparable, in terms of compression ratio and quality of the decoded image, with that of some of the existing methods of fractal image compression.

A different kind of trimming of the search space was described by Jacquin [2]. In his method the reduction takes place in two steps. 1) A "domain pool" consisting of some "domain blocks" is constructed. 2) The domain blocks in the domain pool are classified on the basis of some geometric features.

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 block across the image and taking a constant shift horizontally and vertically. Selecting the domain pool consisting of domain blocks of sizes 16×16 and 8×8, shifts of 4 pixels and 2 pixels respectively are considered. Thus the reduction ratios are 16 and 4 for 8×8 and 4×4 range blocks respectively

Fig. 6: Rose image (256×256, 8bpp): Initial image to start the fractal decoding

for a 256×256 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 Ramamurthi and Gersho [23]. This three class classification scheme is adopted to classify the pool of range blocks as well as the pool of 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 two bits may be required for storing to indicate the class of the range blocks under consideration. In the present encoding algorithm, the range blocks only are classified into two classes which is not only a time saving scheme but it also requires storing of only one bit 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.[26]. 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 HV partitioning method [26], they designed their scheme which results in an increase of compression ratio. The compression ratio (9.97) and the PSNR (31.53) reported by Fisher et al. [26] are almost equal to those reported here.

The search space in the methodology described by Fisher et al.[26] 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.

The basic philosophy of the proposed GA based technique can be adopted to other fractal based coding techniques too. Thomas et al.[30] 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.

To compare the performances of different fractal based image compression schemes, test results of some compression techniques on "Lena" image, as reported are presented in Table 2. The present GA based algorithm has also been implemented on a 512×512, 8 bits/pixel "Lena" image using single level and two level encoding schemes. In single level encoding, range blocks of size 16×16 have been considered. On the other hand, in a two level partitioning scheme, range blocks of sizes 16×16 (parent) and 8×8 (child) have been considered. The compression ratio and PSNR values are found to be 77.25 and 26.13 respectively for single level scheme and 37.16 and 30.87 respectively for two level scheme. A summary of the test results on "Lena" image is shown in Table 3.

Table 2: Test results of some fractal image compression schemes on "Lena" image

Input image	Article	Compression ratio	PSNR (in db)
256×256,6bpp	[2]	8.80	27.70
256×256,8bpp	[26]	9.97	31.53
256×256,8bpp	[35]	16.00	29.10
256×256,8bpp	[46]	10.60	30.72
512×512,8bpp	[85]	40.00	30.20
512×512,8bpp	[41]	44.00	29.10
512×512,8bpp	[30]	41.00	26.56
512×512,8bpp	[38]	53.30	30.30
512×512,8bpp	[73]	44.44	29.10
512×512,8bpp	[26]	36.78	30.71
512×512,8bpp	[34]	69.50	28.30
512×512,8bpp	[61]	65.60	29.90

Table 3: Test results of the proposed GA based compression schemes on "Lena" image

Input image	Type of encoding	Compression ratio	PSNR (in db)
256×256,8bpp	Single level	21.70	26.16
256×256,8bpp	Two level	10.50	30.22
512×512,8bpp	Single level	77.25	26.13
512×512,8bpp	Two level	37.16	30.87

The brief description of the existing methodologies, for finding PIFS code of an image, is provided in Section 2. It has been observed from Table 2 and Table 3 that test results of the proposed GA based fractal image compression scheme are quite satisfactory. The results, in terms of compression ratio and PSNR of the decoded images, are comparable. In fact, the results are, in many cases, better than the results of the existing fractal image compression schemes. Note that, one can manipulate the compression ratio at the cost of quality of the decoded image and the vise versa. In this context, one may also note that in the proposed GA based method for each rough type range block, the number of computations needed in each iteration of GA is of the order of SLb^2, where S is population size, L is string length and $b \times b$ is the size of the range block considered.

The next subsection deals with the results of the magnification technique.

6.2 Image Magnification

The magnification algorithm is implemented on particular portion of several images. Here, only the results of "Lena" image is presented. A portion of the original "Lena" image (Figure 7) is treated as the original input image. The input image is a 128×128, 8 bit/pixel image. The GA based technique, as described, is applied to generate the PIFS codes.

In the case of magnification algorithm, small range blocks of size 2×2 are considered for the computation of PIFS codes. It is true that compression ratio will be reduced by considering smaller range blocks but the finer details of rough type range blocks will be retained. The main aim of a magnification task is to magnify the image keeping all the image details. So, we have considered range blocks of size 2×2. Now, using the obtained PIFS codes, an image of size 256×256 is reconstructed. The reconstructed image is two times magnified than the original image. This image is found to be very close to the original image which is judged by the distortion measure, described in [96]. The PIFS codes are then modified stepwise, as described in [96] to get the images which are 4 times and 8 times magnified than the original one. In each step, the error (distortion measure), in comparison to its previous step, is measured successively.

The proposed algorithm is also compared with the nearest neighbour technique for image magnification in terms of the same distortion measure. Nearest neighbour is probably one of the simplest methods of digital magnification available. Given an image of size $w \times w$, to magnify it by a factor k, every pixel in the new image is assigned the gray value of the pixel in the original image which is nearest to it. This is equivalent to repeating the gray values $k \times k$ times to obtain the magnified image. The resultant image for large magnification factors will have prominent block like structures due to lack of smoothness.

As mentioned above, the magnification algorithm has also been implemented on a part of the "LFA" image, "Seagull" image and "Girl" image. All the images are of size 128×128 and range of gray level values 0 to 255. Other parameters of the algorithm are kept fixed as in the case of "Lena" image. All the results obtained are presented in Table 4.

Table 4: The results obtained in terms of **Distortion** of the Image magnification Algorithm

Input image	Reconstructed Image Statistics								
	MF	Distortion		MF	Distortion		MF	Distortion	
		Fractal	NN		Fractal	NN		Fractal	NN
Lena	2	1.18	1.62	4	1.37	3.11	8	1.24	6.15
LFA	2	2.32	2.37	4	2.85	4.59	8	2.43	9.11
Seagull	2	2.16	2.28	4	2.57	4.41	8	2.48	8.76
Girl	2	2.01	2.42	4	2.51	4.69	8	2.49	9.31

MF=Magnification Factor and NN=Nearest Neighbour

The original input image (part of "Lena") and decoded image are shown in figures 7 and 8 respectively. Figures 9, and 10 are respectively the two times and four times magnified images of the input image using the proposed technique.

Fig. 7: Input image (part of Lena)

Fig. 8: Decoded output image

Fig. 9: Two times magnified image using proposed technique

6.3 Image Edge Extraction

For the specific implementation of the edge extraction algorithm two types of images are considered. A synthetic image ,"Circle" image as shown in Figure 11, and a real life image, "Lena" image, as shown in Figure 4, are treated as the original input images. The "Circle" image is a 128×128, 8 bit/pixel image and on the other hand "Lena" image is a 256×256, 8 bit/pixel image. Here a synthetic "Circle" image is considered so as to judge the performance of the edge extraction algorithm. As there are hardly any measures to judge true edges in a scene. In a synthetic image, the true edges are easily distinguishable from non edges by human eyes. The obtained PIFS code, is then utilized to extract edges.

Fig. 10: Four times magnified image using proposed technique

The obtained PIFS code of “Circle” image is now decoded with the proposed penalty function scheme followed by an edge linking procedure [101]. Other parameters of edge linking algorithm are set experimentally to get the best results. It has been found experimentally that after 20 iterations most of the PIFS codes get stabled at a fixed point. So, we set $N = 20$. Values of other parameters are $T = 0.1$, $L_1 = 3$ and $L_2 = 17$. Obtained decoded “Circle” image and edge image of “Circle” with edge linking are shown in Figure 12 and Figure 13 respectively.

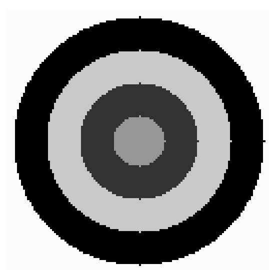


Fig. 11: Original Circle image

PIFS code of “Lena” is also processed similarly. The resultant edge image of “Lena” is shown in Figure 15.

In practice, the result of edge extraction methods are presented as binary images. Usually, an edge pixel is represented by “1” and non edge pixel is by “0”. This process is known as thresholding. To represent the results of the proposed technique in thresholded form, we have set a threshold value “30” and “100” heuristically for “Circle” and “Lena” images respectively. The obtained results of thresholded edge image of “Circle” and “Lena” are shown in Figure 14 and Figure 16 respectively.

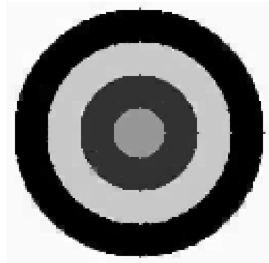


Fig. 12: Decoded Circle image

Fig. 13: Edge image of Circle

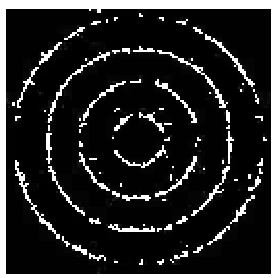


Fig. 14: Thresholded edge image of Circle

Fig. 15: Edge image of Lena

Fig. 16: Thresholded edge image of Lena

7 Discussion and Conclusions

The prime contribution of this article revolves around the PIFS (fractal) codes of images. The PIFS code of an image is generated using a Genetic Algorithm based technique. The performance of the GA based technique in the context of image compression is found to be quite satisfactory. The overall reduction in the search space is found to be of the order of 20 times. The compression ratios and the quality of the decoded images are found to be comparable with that of the other existing fractal based methods.

The factors on which the performance of the GA based PIFS technique hinges upon is the proper selection of the parameter values in GA *viz.*, (i) the size of the initial population, (ii) the number of iterations performed.

Another important factor that influences the performance of the proposed technique is the selection of fitness function of strings in the GA. It indicates the proper selection of the approximations of the range blocks in the proposed algorithm. The overall effect of it can be visualized in the quality of the decoded image. Root mean square error (RMSE) is used as the fitness function for the set of strings. 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 RMSE in the proposed method.

The article also discusses some other image processing algorithms such as image magnification and image edge extraction using PIFS code. These investigations regarding the applicability of the fractal technique, in particular PIFS based technique, to other possible areas of image processing are unique of their kind. Very few contributions have been made in this area. The most important feature of the proposed techniques of fractal image magnification and fractal image edge extraction is that these utilize the coded (fractal) version of the input image instead of the original image.

These proposed schemes are highly dependent on the compression scheme adopted. There are several compression schemes available in the literature. But the proposed schemes are applicable only on the PIFS code. Any other code obtained by other compression schemes, like run length coding, vector quantization coding or Wavelet based images coding, can not be used as input for the proposed schemes.

The sizes of the range blocks considered play a vital role in image compression, magnification and edge extraction using fractal technique. The last two algorithms are in opposite direction of the first one from the point of view of range block size. In particular finer image detail can be retained by using range blocks of smaller size and more compression can be achieved through range blocks of larger size. While performing the magnification and the edge extraction task it is very important to use the information regarding finer image detail. So, one can think of an optimal range block size for which good quality magnified images and edge images can be reconstructed from the fractal codes and at the same time considerable amount of compression can be achieved.

Generation of PIFS code is a block coding method. But, as it tries to approximate image regions, the block effect is reduced to a great extent. Regarding this, it should be noted that the advantage of fractal image magnification is that it magnifies the image by expanding the fractal codes which are independent of the image resolution. The scheme can be expressed as an approximation resulting in a sharper expanded image. Other image magnification schemes using pixel replication make magnified images blocky, blurry and patchy after a certain extent of expansion.

On the other hand, it has been found visually that there are some differences in the edges extracted by the proposed edge extraction scheme on the compressed images with the Canny operator on the original images. The edges extracted by the proposed scheme are smudgy where as the edges found by the Canny operator [15] are smooth. Otherwise it may be observed that, the edges obtained by the Canny operator are more or less present in the edge images of the proposed scheme.

The proposed edge extraction technique is noise sensitive while most of the edge detection algorithms are noise insensitive. But this problem can not be avoided as the coded version of the image is used instead of the original one. If noise is already there in the original image then the coded version will usually inherit that noise and hence it will be reflected in the edge image obtained by the present algorithm. This problem can be avoided in two ways. One can adopt an encoding mechanism which is noise insensitive or apply a noise removal algorithm before doing the encoding. The problem of noise sensitivity is a general problem in the field of image processing in compressed domain.

Finally, it can be concluded that inspite of a little difficulties, PIFS codes can be successfully used as an input to the image processing system. The present work is unique of its kind as it developed a new and improved image processing system where processing can be carried out with the coded version of the image instead of the original one resulting in reduction in the memory requirements as well as reduction in the computational cost.

References

[1] M. F. Barnsley, *Fractals Everywhere*. New York: Academic Press, 1988.

[2] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, pp. 18–30, 1992.

[3] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Fractal based image coding using genetic algorithm," in *Pattern Recognition, Image Processing and Computer Vision. Recent Advances* (P. P. Das and B. N. Chatterji, eds.), pp. 86–91, New Delhi: Narosa Publishing House, 1995.

[4] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Technique for fractal image compression using genetic algorithm," *IEEE Transactions on Image Processing*, vol. 7, pp. 586–593, 1998.

[5] D. E. Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning.* Reading: Addison - Wesley, 1989.

[6] L. Davis, *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold, 1991.

[7] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs.* Berlin: Springer Verlag, 1992.

[8] P. S. Chavez(Jr.), "Digital merging of landsat TM and digitised NHAP data for 1:24000 scale image mapping," *Photogrammetric Engineering and Remote Sensing*, vol. 52, pp. 1637–1646, 1986.

[9] R. Welch and M. Ehlers, "Merging multi resolutions SPOT HRV and Landsat TM data," *Photogrammetric Engineering and Remote Sensing*, vol. 53, pp. 301–303, 1987.

[10] R. C. Gonzalez and R. R. Wood, *Digital Image Processing.* Reading: Addison- Wesley, 1993.

[11] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, pp. 1153–1160, 1981.

[12] S. K. Park and R. A. Schowengerdt, "Image reconstruction using parametric cubic convolution," *Computer Vision, Graphics, and Image Processing*, vol. 23, pp. 258–272, 1983.

[13] R. A. Schowengerdt, S. K. Park, and R. Gray, "Topics in two dimensional sampling and reconstruction of images," *International Journal of Remote Sensing*, vol. 5, pp. 333–347, 1984.

[14] W. K. Pratt, *Digital Image Processing.* New York: John Wiley and Sons Inc., 1991.

[15] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[16] D. Marr and E. C. Hilderth, "Theory of edge detection," in *Proceedings of Royal Society, Series B*, (London, UK), pp. 187–217, 1980.

[17] R. M. Haralick, "Digital step edge from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 58–86, 1984.

[18] A. E. Jacquin, *Fractal Theory of Iterated Markov Operators With Applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, Atlanta, USA, 1989.

[19] A. E. Jacquin, "A novel fractal block-coding technique for digital images," in *Proceedings of International Conference on Acoustics, Speach and Signal Processing (ICASSP'90)*, (Alberquerque, USA), pp. 2225–2228, 1990.

[20] F. Dudbridge, *Image Approximation by Self Affine Fractals.* PhD thesis, Imperial College, London, UK, 1992.

[21] Y. Fisher, *Fractal Image Compression: Theory and Application.* New York: Springer Verlag, 1995.

[22] Y. Fisher, *Fractal Image Encoding and Analysis. NATO ASI Series F, vol. 159.* New York: Springer Verlag, 1998.

[23] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communications*, vol. COM-34, pp. 1105–1115, 1986.

[24] A. D. Sloan, "Low-bit-rate fractal image coding," in *Visual Information Processing III, Volume 2239 of SPIE Proceedings* (F. O. Huck and R. D. Juday, eds.), pp. 210–213, 1994.

[25] H. Lin and A. N. Venetsanopoulos, "Perceptually lossless fractal image compression," in *Visual Communication and Image Processing, Volume 2727 of SPIE Proceedings* (R. Ansari and M. J. Smith, eds.), pp. 1394–1399, 1996.

[26] Y. Fisher, E. W. Jacbos, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression* (J. A. Storer, ed.), pp. 35–61, Kluwer Academic Publishers, 1992.

[27] D. Saupe and S. Jacob, "Variance based quadtree in fractal image compression," *Electronics Letters*, vol. 33, pp. 46–48, 1997.

[28] F. Davoine, J. Sevesson, and J. M. Chessary, "A mixed triangular and quadrilateral partition for fractal image coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 284–287, 1995.

[29] H. L. Ho and W. K. Cham, "Attractor image coding using lapped partitioned iterated function systems," in *Proceedings of IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2917–2920, 1997.

[30] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Transactions on Image Processing*, vol. 4, pp. 832–838, 1995.

[31] M. H. Loew, D. Li, and R. L. Pickholtz, "Adaptive PIFS model in fractal image compression," in *Medical Imaging 1996: Image Display, Volume 2707 of SPIE Proceedings* (Y. Kim, ed.), pp. 284–293, 1996.

[32] M. Tanimoto, H. Ohyama, and T. Kimoto, "A new fractal image coding scheme employing blocks of variable shapes," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 137–140, 1996.

[33] D. Saupe and M. Ruhl, "Evolutionary fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 161–164, 1996.

[34] M. Ruhl, H. Hartenstein, and D. Saupe, "Adaptive partitionings for fractal image compression," in *IEEE International Conference on Image Processing (ICIP'97)*, (Santa Barbara, USA), 1997.

[35] E. Reusens, "Overlapped adaptive partitioning for image coding based on the theory of iterated function systems," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, (Adelaide, Australia), pp. 569–572, 1994.

[36] J. S. Chen, "Fractal image compression based on visual perception," in *Human Visiom, Visual Processing and Digital Display VI, Volume 2411 of SPIE Proceedings* (B. E. Rogowitz and J. P. Allebach, eds.), pp. 92–99, 1995.

[37] K. U. Barthel, "Entropy constrained fractal image coding," *Fractals*, vol. 5, pp. 17–26, 1997.

[38] R. Rinaldo and G. Calvango, "Image coding by block prediction of multi-resolution subimages," *IEEE Transactions on Image Processing*, vol. 4, pp. 909–920, 1995.

[39] M. G. Alkhansari and T. S. Huang, "Fractal-based techniques for a generalised image coding method," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 122–126, 1994.

[40] D. Saupe, "Fractal image compression via nearest neighbor search," in *NATO ASI on Fractal Image Encoding and Analysis*, (Trondheim, Norway), 1995.

[41] H. Lin and A. N. Venetsanopoulos, "A pyramid algorithm for fast fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 596–599, 1995.

[42] B. Simon, "A pyramid algorithm for fast fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 278–281, 1995.

[43] L. Lepsoy and G. E. Oien, "Fast attractor image encoding by adaptive codebook clustering," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 177–197, New York: Springer Verlag, 1995.

[44] C. J. Wein and I. F. Blake, "On the performance of fractal compression with clustering," *IEEE Transactions on Image Processing*, vol. 5, pp. 522–526, 1996.

[45] J. Kominek, "Codebook reduction in fractal image compression," in *Still Image Compression II, Volume 2669 of SPIE Proceedings* (R. L. stevenson, A. I. Drukarev, and T. R. Gardos, eds.), pp. 33–41, 1995.

[46] M. E. Haziti, H. Cherifi, and D. Aboutajdine, "Complexity reduction in fractal image compression," in *Proceedings of the IASTED International Conference on Signal and Image Processing(SIP'97)*, (New Orleans, USA), pp. 245–250, 1997.

[47] J. Signes, "Reducing the codebook size in fractal image compression by geometrical analysis," in *Visual Communication and Image Processing, Volume 2727 of SPIE Proceedings* (R. Ansari and M. J. Smith, eds.), pp. 1400–1409, 1996.

[48] B. E. Wohlberg and G. de Jager, "On reduction of fractal image compression encoding time," in *IEEE south African symposium on Communications and Signal processing (COMSIG'94)*, (University of Stellenbosch), pp. 158–161, 1994.

[49] Z. Baharav, D. Malah, and E. Karnin, "Hierarchical interpretation of fractal image coding and its application to fast decoding," in *Proceedings of the IEEE International Conference on Digital Signal Processing*, (Nicosia, Cyprus), pp. 190–195, 1993.

[50] B. Bani-Eqbal, "Speeding up fractal image compression," in *Still Image Compression, Volume 2418, SPIE Proceedings* (M. Rabbani, E. J. Delp, and S. A. Rajala, eds.), pp. 67–74, San Jose, USA, 1995.

[51] K. Belloulata, A. Baskurt, and R. Prost, "Fast directional fractal coding of subbands using decision directed clustering for block classification," in *Proceedings of IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 3121–3124, 1997.

[52] G. Caso, P. Obrador, and C. C. J. Kuo, "Fast method for fractal image encoding," in *Visual Communication and Image Processing, Volume 2501 of SPIE Proceedings* (L. T. Wu, ed.), pp. 583–594, Taipei, Taiwan, 1996.

[53] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images," in *Video Communications and PACS for Medical Applications, Volume 1977 of SPIE Proceedings* (R. A. Mattheus, A. J. Duerinckx, and P. J. V. Otterloo, eds.), pp. 397–408, 1993.

[54] B. S. Everitt, "Unresolved problems in cluster analysis," *Biometrics*, vol. 35, pp. 169–181, 1979.

[55] W. Skarbek, "On convergence of affine fractal operators," *Image Processing and Communications*, vol. 1, pp. 33–41, 1995.

[56] W. Skarbek, "Analysis of fractal operator convergence by graph method," *Fundamenta Informaticae*, vol. 34, pp. 429–440, 1998.

[57] M. G. Alkhansari and T. S. Huang, "A system/graph theoretical analysis of attractor coders," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2705–2708, 1997.

[58] B. Hurtgen and S. F. Simon, "On the problem of convergence in fractal coding schemes," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 103–106, 1994.

[59] B. Hurtgen and T. Hain, "On the convergence of fractal transforms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, (Adelaide, Australia), pp. 561–564, 1994.

[60] G. Davis, "Self quantized wavelet subtree: A wavelet based theory fractal image compression," in *Wavelet Application II, Volume 2491 of SPIE Proceedings* (H. H. Szu, ed.), pp. 141–152, 1995.

[61] G. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Transactions on Image Processing*, vol. 7, pp. 141–154, 1998.

[62] G. Davis, "Why fractal block coders work," in *Fractal Image Encoding and Analysis, NATO ASI Series F, vol. 159* (Y. Fisher, ed.), pp. 3–19, New York: Springer Verlag, 1998.

[63] M. Dekking, "Fractal image coding: Some mathematical remarks on its limits and its prospects," in *Fractal Image Encoding and Analysis, NATO ASI Series F, vol. 159* (Y. Fisher, ed.), pp. 117–131, New York: Springer Verlag, 1998.

[64] J. H. Elton, "An ergodic theorem for iterated maps," *Ergodic theory and Dynamical Systems*, vol. 7, pp. 481–488, 1987.

[65] B. Forte and E. R. Vrscay, "Solving the inverse problem function and image approximation using iterated function systems I: Theoretical basis," *Fractals*, vol. 3, pp. 325–334, 1994.

[66] D. M. Monro, "Generalized fractal transform: Complexity issues," in *Proceedings of IEEE Data Compression Conference, (DCC'93)* (J. A. Storer and M. Cohn, eds.), pp. 254–261, 1993.

[67] G. E. Oien, Z. Baharav, S. Lepsoy, E. Karnin, and D. Malah, "A new improved collage theorem with application to multiresolution fractal image coding," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, pp. 565–568, 1994.

[68] K. Kim and R. H. Park, "Image coding based on fractal approximation and vector quantization," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 132–136, 1994.

[69] K. Kim and R. H. Park, "Still image coding based on vector quantization and fractal approximation," *IEEE Transactions on Image Processing*, vol. 4, pp. 587–597, 1996.

[70] A. Bogdan and H. Meadows, "Kohonen neural network for image coding based on iteration transform theory," in *Neural and Stochastic Methods in Image and Signal Processing* (S. S. Chen, ed.), pp. 425–436, Volume 1766 of SPIE Proceedings, 1992.

[71] W. Skarbek, "Image compression using pixel neural network," in *Soft-computing for Image Processing* (S. K. Pal, A. Ghosh, and M. K. Kundu, eds.), Heidelberg: Physica Verlag, 1999.

[72] A. V. de Walle, "Merging fractal image compression and wavelet transform methods," in *NATO ASI on Fractal Image Encoding and Analysis*, (Trondheim, Norway), 1995.

[73] N. T. Thao, "A hybrid fractal-DCT coding scheme for image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 169–172, 1996.

[74] B. E. Wohlberg and G. de Jager, "Fast image domain fractal compression by DCT domain block matching," *Electronics Letters*, vol. 31, pp. 869–870, 1995.

[75] K. U. Barthel, J. Schuttemeyer, T. Voyi, and P. Noll, "A new image coding technique unifying fractal and transform coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 112–116, 1994.

[76] D. M. Monro, "A hybrid fractal transform," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'93)*, pp. 169–172, 1993.

[77] V. Ratnakar, E. Feig, and P. Tiwari, "Fractal based hybrid compression scheme," in *Visual Communication and Image processing, Volume 2308 of SPIE Proceedings* (A. K. Katsaggelos, ed.), pp. 448–454, 1994.

[78] P. D. Wakefield, D. M. Bethel, and D. M. Monro, "Hybrid image compression with implicit fractal terms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2933–2936, 1997.

[79] N. Zhang and H. Yan, "Hybrid image compression method based on fractal geometry," *Electronics Letters*, vol. 27, pp. 406–408, 1991.

[80] J. Domaszewich, S. Kullinski, and V. A. Vaishampayan, "Fractal coding versus classified transform coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96*, (Lausanne, Switzerland), pp. 149–152, 1996.

[81] Y. Fisher, T. P. Shen, and D. Rogovin, "Comparison of fractal methods with discrete cosine transform (DCT) and wavelets," in *Neural and Stochastic Methods in Image and Signal Processing III, Volume 2308, of SPIE Proceedings* (S. S. Chen, ed.), pp. 132–143, 1994.

[82] D. W. Lin and R. H. Park, "Fractal image coding as generalised predictive coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 117–121, 1994.

[83] J. Scharinger, F. Pichler, H. G. Feichtinger, and F. Leberl, "Comparison of lossy image compression techniques with respect to their impact on edge detection," in *Application of Digital Image Processing XIX, Vol. 2847 of SPIE Proceedings*, (Denver, USA), pp. 479–490, 1996.

[84] F. Dudbridge, "Least-squares block coding by fractal functions," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 229–241, New York: Springer Verlag, 1995.

[85] H. Lin and A. N. Venetsanopoulos, "Incorporating nonlinear contractive functions into the fractal coding," in *Proceedings of IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, (Seoul, Korea), pp. 169–172, 1994.

[86] G. E. Oien and G. Narstad, "Fractal compression of ECG signals," in *Fractal Image Encoding and Analysis* (Y. Fisher, ed.), pp. 201–226, New York: Springer Verlag, 1998.

[87] G. Vines, "Orthogonal basis of IFS," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 199–214, New York: Springer Verlag, 1995.

[88] C. S. Kim, R. C. Kim, and S. U. Lee, "Novel fractal image compression with non iterative decoder," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 268–271, 1995.

[89] R. Hamzaoui, "A new decoding algorithm for fractal image compression," *Electronics Letters*, vol. 14, pp. 1273–1274, 1996.

[90] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression.* Wellesley: AK Peters Ltd., 1993.

[91] S. K. Mitra and C. A. Murthy, "Mathematical framework to show the existence of the attractor of the partitioned iterative function systems," *Pattern Recognition*, vol. 33, 2000.

[92] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 731–747, 1996.

[93] S. Bandyopadhyay, *Pattern Classification Using Genetic Algorithm.* PhD thesis, Indian Statistical Institute, Calcutta, India, 1998.

[94] S. K. Pal, D. Bhandari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognition Letters*, vol. 15, pp. 261–271, 1994.

[95] N. R. Pal, S. Nandi, and M. K. Kundu, "Self crossover - A new genetic operator and its application to feature seclection," *International Journal of Systems Secince*, vol. 29, pp. 207–212, 1998.

[96] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "A technique for image magnification using partitioned iterative function system," *Pattern Recognition*, vol. 33, 2000.

[97] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Digital image magnification using fractal operators and genetic algorithm," in *Computational Intelligence and Applications* (P. S. Szczepaniak, ed.), pp. 250–259, Heidelberg: Physica Verlag, 1998.

[98] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "A study on partitioned iterative function systems for image compression," *Fundamenta Informaticae*, vol. 34, pp. 413–428, 1998.

[99] S. K. Pal and D. D. Majumder, *Fuzzy Mathematical Approach to Pattern Recognition.* New York: John Wiley (Halsted Press), 1986.

[100] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithm," *Pattern Recognition Letters*, vol. 17, pp. 825–832, 1996.

[101] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Edge extraction in the compressed domain using fractal reconstruction." *IEEE Transactions on Image processing*, (Communicated).