

A New Probabilistic Approach for Fractal Based Image Compression

Suman K. Mitra

Dhirubhai Ambani Institute of Information and Communication Technology

Near Indroda Circle, Gandhinagar 382 007, Gujarat, India

suman_mitra@daiict.ac.in

Malay K. Kundu*, **C.A. Murthy**, **Bhargab B. Bhattacharya**

Indian Statistical Institute

203, B. T. Road, Calcutta 700108, India

{malay,murthy,bhargab}@isical.ac.in

Tinku Acharya

Avisere, Chandler, AZ 85226, USA

tinku.acharya@ieee.org

Abstract. Approximation of an image by the attractor evolved through iterations of a set of contractive maps is usually known as fractal image compression. The set of maps is called iterated function system (IFS). Several algorithms, with different motivations, have been suggested towards the solution of this problem. But, so far, the theory of IFS with probabilities, in the context of image compression, has not been explored much. In the present article we have proposed a new technique of fractal image compression using the theory of IFS and probabilities. In our proposed algorithm, we have used a multiscaling division of the given image up to a predetermined level or up to that level at which no further division is required. At each level, the maps and the corresponding probabilities are computed using the gray value information contained in that image level and in the image level higher to that level. A fine tuning of the algorithm is still to be done. But, the most interesting part of the proposed technique is its extreme fastness in image encoding. It can be looked upon as one of the solutions to the problem of huge computational cost for obtaining fractal code of images.

Keywords: Iterated function System (IFS), Collage theorem, measure, Markov operator, image compression.

*Address for correspondence: Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Calcutta 700108, India.

1. Introduction

A set of contractive maps on a space is called iterated function system (IFS) on the same space. The most interesting property of IFS is that it produces a fixed point when the maps are used recursively starting with any arbitrary set. The fixed point is called the attractor of the IFS. The theory of image coding using IFS was first proposed by Barnsley [2]. He modeled real life images by the attractor evolved through iterations of an IFS. With the help of iterated function system, along with the Collage theorem, Barnsley laid the foundation stone of fractal image compression. However, in most of his published research papers, implementation part of the theory of IFS to real life images has not been very clear as the demonstrations are mostly restricted to binary images. Seemingly, a photocopying machine has been designed by means of which coefficients of maps are computed [3]. The concept of photocopying machine has also been extended to the case of gray scale images [3]. In this case, the machine is computing not only the coefficients of maps but also the probability values associated with each map. The reason behind using the IFS with probabilities is that a Markov operator associated with the probability measure whose support is the support of the given image can be defined. It has also been shown that the Markov operator is a contractive map on the space of all probability measures. Barnsley also presented the Collage theorem for measure [3] for which the Markov operator possesses an invariant measure in probability space. From this Collage theorem it has been found that an IFS with probabilities can produce an invariant measure such that the support of the invariant measure is the unique fixed point of the IFS concerned.

The first ever published algorithm of fractal image compression was suggested by Jacquin [12]. This scheme is based on partitioned or local IFS [13]. A variety of algorithms have been explored [8, 6, 16, 7, 14] there after. Most of these schemes are closely related to the technique described by Jacquin. Unfortunately, the theory of IFS with probabilities, in the context of image compression, has not been explored much. Though the theoretical results are existing in the literature for a long time [3, 11, 5], there are very few attempts to implement the theory on computer for images. It may be due to the fact that some of the mathematical terms need to be properly explained and interpreted in the context of computer programme on a digital images apart from considering all other implementation details.

An algorithm, which is not in the line of PIFS based fractal image compression has been suggested here. We have used a multiscaling division of the given image. At each level the coefficients of maps and the corresponding probabilities are computed using the gray value information. This is an attempt to implement the theory of IFS with probabilities for digital images. There are very few references [9, 10] in the literature on fractal image compression, besides the work done by Barnsley [3], on this particular aspect. The present algorithm is appeared to be extremely fast for encoding a given image. A comparison of the proposed method with a GA based fractal method [14] is provided. It is found that the present scheme is at least 300 times faster than that of the GA based based methodology. Note that, previously it has been found that GA based fractal scheme is faster than the conventional fractal scheme which uses exhaustive search.

The algorithm described here utilizes the probabilistic relation exists in between pixel values of a given image. Contribution of a particular pixel, in terms of intensity (gray level value), towards the total intensity of the image is computed under a multiscaling partition. If ' p ' is the proportion of the total intensity carried by a pixel then one can express the intensity of that pixel (say ' g ') as $g = p \times G$, where ' G ' is the sum of all pixel values. In this algorithm, ' p ' is computed as a product of a set of probability values which were obtained from the image information (pixel values) after braking the image

in multiscaling order. In the reconstruction process each pixel value is again regenerated using ‘ G ’ and the stored set of probability values. Note that ‘ G ’ is also stored within the code. The scheme performs very fast as no search is performed for finding appropriate domain blocks and maps for a set of range blocks. This sort of searching is very common in case of PIFS based image compression. In the present case, the range blocks are the image partitions at a particular subdivision and the domain blocks for these range blocks are the image partitions at immediate higher level subdivision. The same kind of partitioning is also utilized by Dudbridge [4]. As far as we understand, the present algorithm is only similar with the algorithm suggested by Dudbridge [4] in case of selection of domain blocks and range blocks under multiscaling partitioning. The procedure for computation of codes is not exactly the same as that of the method suggested by Dudbridge. Note that no computation such as ‘least square’ is adopted in our method.

There is a lot of scope for improving the present algorithm towards the efficiency in the sense of compression ratio as well as quality of the decoded image. It is obvious that a fine tuning of the scheme is needed. Work, in this regard, is in progress. Our main objective, in this article, is to investigate the art of finding a probabilistic approach for approximating the given image. The high speed of execution of the method appeared to be the main advantage.

The mathematical foundation of IFS with probabilities is outlined in Section 2. The methodology of the proposed probabilistic algorithm is described in Section 3. Section 4 presents implementation and the results. Discussion and conclusions are provided in Section 5.

2. Mathematical Foundation

The detailed mathematical description is given in [2, 3]. Some relevant definitions and theorems are stated here. Proofs of the theorems have not been stated as all these are already given in [2]. Starting with the definition of complete metric space, IFS has been defined in the space of probability measures. Invariant measure of IFS in terms of Collage theorem has also been described in the same space.

The most important result which is being used here, is the theorem of invariant measure of IFS with probabilities defined on a probability space. Other definitions and theorems are stated for better understanding of the theorem of interest. To define the IFS with probabilities in the probability space, we need to define, the probability space and the IFS in that space. Again we need to define the probability measure and the contractive mapping theorem on this space. Let us start with complete metric space.

Definition 1: Let (X, d) be a complete metric space. Then $\mathcal{H}(X)$ denotes the space of all nonempty compact subsets of X . ♠

Definition 2: Let The map $w : X \rightarrow X$ is called a contractive map with contractivity factor s with $0 \leq s < 1$ if

$$d(w(x), w(y)) \leq s d(x, y) \quad \forall x, y \in X. \spadesuit$$

Lemma 1: Let w be a contractive map on the complete metric space (X, d) . Then w maps $\mathcal{H}(X)$ into itself where

$$w(B) = \{w(x) : x \in B\}, \quad \forall B \in \mathcal{H}(X).$$

w is a contractive map on $(\mathcal{H}(X), h)$ with contractivity factor s . Here ‘ h ’ is called Hausdorff metric. ♠

Lemma 2: Let (X, d) be a complete metric space. Let $\{w_i : i = 1, 2, \dots, n\}$ be contractive map on $(\mathcal{H}(X), h)$. Let the contractivity factor for w_i be denoted by s_i for each i . Define $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ by

$$W(B) = \bigcup_{i=1}^n w_i(B), \quad \forall B \in \mathcal{H}(X).$$

Then W is a contractive map with contractivity factor $s = \max\{s_i : i = 1, 2, \dots, n\}$. ♠

Definition 3: An iterated function system (IFS) consists of a complete metric space (X, d) together with a finite set of contractive maps $w_i : X \rightarrow X$, with respective contractivity factors s_i , for $i = 1, 2, \dots, n$. The notation for this IFS is $\{X; w_1, w_2, \dots, w_n\}$ and its contractivity factor is s where $s = \max\{s_1, s_2, \dots, s_n\}$. ♠

The preceding results are useful in proving the following theorem of the existence of a fixed point of IFS.

Theorem 1 (IFS Theorem): Let $\{X; w_1, w_2, \dots, w_n\}$ be an IFS with contractivity factor s . Then the transformation $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ defined by

$$W(B) = \bigcup_{i=1}^n w_i(B), \quad \forall B \in \mathcal{H}(X),$$

is a contractive map on the complete metric space $(\mathcal{H}(X), h)$ with contractivity factor s ; i.e.

$$h(W(B), W(C)) \leq s \cdot h(B, C), \quad \forall B, C \in \mathcal{H}(X).$$

Also $W^N(B)$ is defined as

$$W^N(B) = W(W^{N-1}(B)); \quad \forall B \in \mathcal{H}(X) \text{ and } \forall N \geq 2,$$

where

$$W^1(B) = W(B); \quad \forall B \in \mathcal{H}(X).$$

It has a unique fixed point, called **attractor**, $A \in \mathcal{H}(X)$, which obeys

$$A = W(A) = \bigcup_{i=1}^n w_i(A),$$

and is given by

$$A = \lim_{N \rightarrow \infty} W^N(B) \quad \forall B \in \mathcal{H}(X). \quad \spadesuit$$

Besides the contractive maps, there may exist a condensation map. The theorem of IFS is also valid when the condensation map is included in the IFS along with the contractive maps.

Definition 4: Let (X, d) be a complete metric space and let $C \in \mathcal{H}(X)$. Define a transformation $w_0 : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ by $w_0(B) = C; \forall B \in \mathcal{H}(X)$. Then w_0 is called a condensation transformation and C is called the associated condensation set. ♠

The condensation map is also a contractive map with unique fixed point. Next we are going to state the Collage theorem of IFS.

Theorem 2 (Collage theorem): Let (X, d) be a complete metric space. Let $T \in \mathcal{H}(X)$ and let $\epsilon \geq 0$ be given. Choose an IFS $\{X; w_1, w_2, \dots, w_n\}$ with contractivity factor s such that

$$h(T, \bigcup_{i=1}^n w_i(T)) \leq \epsilon,$$

where h is the Hausdorff metric. Then

$$h(T, A) \leq \frac{\epsilon}{1-s},$$

where A is the attractor of IFS. ♠

Now we shall define IFS defined on the space of all probability measures. In particular, the Markov operator, which is a contractive map on the space of all probability measures, is defined below. For this purpose, one needs to first define a metric on the space of all probability measures.

Definition 5: An iterated function system with probabilities consists of an IFS $\{X; w_1, w_2, \dots, w_N\}$ together with an ordered set of numbers $\{p_1, p_2, \dots, p_N\}$, such that

$$p_1 + p_2 + \dots + p_N = 1 \text{ and } p_i > 0 \forall i.$$

The probability p_i is associated with the transformation w_i . ♠

Definition 6: Let $\square = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b, c \leq y \leq d\}$,

where $a < b$ and $c < d$ are real constants. Also let \mathcal{P} denote the set of all probability measures on \square . The Hutchinson metric d_H on \mathcal{P} is defined by

$$d_H(\mu, \nu) = \sup_{\square} \left\{ \int_{\square} f d\mu - \int_{\square} f d\nu : f : \square \rightarrow \mathbb{R} \text{ is a continuous function} \right.$$

$$\left. \text{and obeys } |f(x) - f(y)| \leq d(x, y) \forall x, y \in \square, \forall \mu, \nu \in \mathcal{P}. \spadesuit \right.$$

Theorem 3: Let \mathcal{P} denote the set of all probability measures on \square and let d_H denote the Hutchinson metric. Then (\mathcal{P}, d_H) is a complete metric space. ♠

Definition 7: Let $\{\square; w_1, w_2, \dots, w_n; p_1, p_2, \dots, p_n\}$ be an IFS with probabilities. The Markov operator associated with the IFS is the function $M : \mathcal{P} \rightarrow \mathcal{P}$ defined by

$$M(\nu) = p_1 \nu \circ w_1^{-1} + p_2 \nu \circ w_2^{-1} + \dots + p_n \nu \circ w_n^{-1}; \quad \forall \nu \in \mathcal{P}.$$

$$\text{i.e. } M(\nu) = \sum_{i=1}^n p_i \nu \circ w_i^{-1}. \spadesuit$$

The preceding definitions and results are needed to prove the following theorem.

Theorem 4 (Hutchinson's Theorem): Let $M : \mathcal{P} \rightarrow \mathcal{P}$ be the Markov operator associated with an IFS with probabilities, where each transformation has contractivity factor $0 \leq s < 1$. Then M is a contractive map, with contractivity factor s , with respect to the Hutchinson metric \mathcal{P} ; i.e.,

$$d_H(M(\nu), M(\mu)) \leq s d_H(\nu, \mu); \quad \forall \nu, \mu \in \mathcal{P}.$$

In particular, there is a unique measure $\mu \in \mathcal{P}$ such that $M\mu = \mu$. Also, if $M^N(\nu) = M(M^{N-1}(\nu))$ for $\nu \in \mathcal{P}$, $N \geq 2$, then

$$\lim_{N \rightarrow \infty} M^N(\nu) = \mu, \quad \forall \nu \in \mathcal{P}.$$

where the convergence is with respect to the Hutchinson metric on \mathcal{P} . ♠

The Collage theorem for measures is stated below. The theorem states that any given probability measure is approximated by the fixed point (unique invariant measure) of an IFS with probabilities.

Theorem 5: Let $\{\square; w_1, w_2, \dots, w_n; p_1, p_2, \dots, p_n\}$ be an IFS with probabilities. Let μ be the associated invariant measure. Then the support of μ is the attractor of the IFS $\{\square; w_1, w_2, \dots, w_n\}$. ♠

Theorem 6 (Collage Theorem for Measure): Let $\{\square; w_1, w_2, \dots, w_n; p_1, p_2, \dots, p_n\}$ be an IFS with probabilities. Let μ be the associated invariant measure. Let $s \in (0, 1)$ be the contractivity factor for the IFS. Let $M : \mathcal{P} \rightarrow \mathcal{P}$ be the associated Markov operator. Let $\nu \in \mathcal{P}$. Then

$$d_H(\nu, \mu) \leq \frac{d_H(\nu, M(\nu))}{(1-s)}. \quad \spadesuit$$

Note that the theory of IFS with probabilities can be extended to include a *condensation measure*. Once IFS with probabilities or the Markov operator associated with ν is at hand, one can approximate the measure ν by the invariant measure μ starting from any measure on the same space. In the context of image compression, the process of computing the Markov operator is called the encoding process. Likewise the process of finding the invariant measure μ , hence the attractor of the concerned IFS, starting from any initial measure, is called the process of decoding.

To implement the aforesaid theory on real life images the problem is to find a way of defining probability measure corresponding to an image. It is extremely difficult to define the probability measure of the image. However an algorithm called gray scale photocopying algorithm, seemingly, was suggested by Barnsley [3] as a part of his patented work. We don't know the details of the patented work. A description of the algorithm is given in [3]. But the implementation part is not clear from this description. The methodology of the present algorithm is motivated by this description. We are presenting here only the salient features of the algorithm suggested by Barnsley [3].

The gray scale photocopying algorithm is a procedure for computing the invariant measure μ , the unique solution of $Q(\mu) = \mu$, $\mu \in \mathcal{P}$. Consider a gray scale photograph supported on \square . If light is passed through the photograph, it reflects a certain amount of light. Dark regions reflect relatively less light, while bright regions reflect relatively large. A function ν is introduced which describes the amount of light reflected from different regions of the photograph. The photograph is now represented by $\nu \in \mathcal{P}$, where \mathcal{P} is the set of all normalized Borel measures on \square . The photocopying machine consists of some lenses and filters. The emitted light, from different parts of the given image, is controlled by system of lenses which apply the affine transformations w_i ; $i = 1, 2, \dots, n$. Each lens system has its own filter p_i , $i = 1, 2, \dots, n$. A filter attenuates the light that passes through it by a factor proportional to the number p_i . The output of the machine is taken on a photographic plate. The resulting image is $Q(\nu)$.

The basic idea of the machine is adjusting the affine maps w_i and the probabilities p_i one by one starting from all set at zero. For example, first of all, the transformation w_1 and the probability p_1 are adjusted to make the output image as good an approximation as possible to a part of the target measure ν . The adjustment of w_1 and p_1 are such that $p_1\nu \circ w_1^{-1}(B) < \nu(B)$ for every Borel subset B of \square . Thus

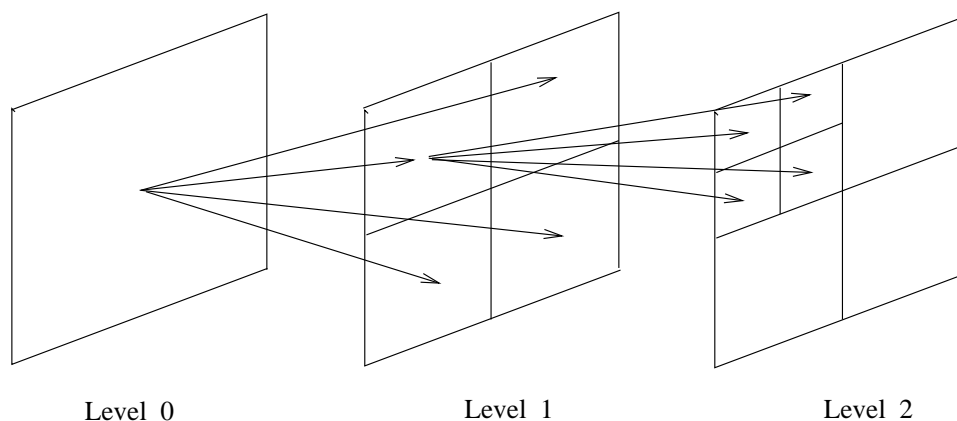


Figure 1. Multiscaling type division of an image

$(\nu - p_1\nu \circ w_1^{-1})$ is a Borel measure. Next the affine transformation w_2 and probability p_2 are adjusted in such a way that $p_1\nu \circ w_1^{-1} + p_2\nu \circ w_2^{-1}$ is as good an approximation as possible of a larger part of ν . Again the adjustment is such that $\nu - (p_1\nu \circ w_1^{-1} + p_2\nu \circ w_2^{-1})$ is a Borel measure. New transformations and probabilities are added successively to the IFS with probabilities to improve the approximation of ν by $Q(\nu)$. At all stages the contractivity of the Markov operator is maintained to be s by insisting that each affine map has a contractivity factor less than s , where $0 \leq s < 1$. Once the output image looks sufficiently like the input image, the transformations and the corresponding probabilities are recorded.

In the next section we have described our proposed algorithm.

3. Proposed Probabilistic Approach for Image Encoding

In the present work we have used a multiscaling division of the given image as described in figure 1. At each level of partitioning, the image or subimage is divided into four quadrants. Each quadrant is called the child of its parent image or subimage. The Markov operator is computed and the approximation of the image by means of the invariant measure of the Markov operator is carried out at each level of partitioning. The process of partitioning the image, or the subimage as the case is, and computing the Markov operator are performed up to a predetermined level or up to a level at which no further division is required. No further division is required at any level indicates that the successful approximation of the given image through the obtained Markov operator. To compute the Markov operator, it is essential to find the maps and their corresponding probabilities. At each level, probabilities are computed by the ratio of sum of gray values of a child subimage to the sum of gray values of its parent subimage. This conforms the sum of four probabilities, at each level, to be unity. The ratio of the gray level value of a pixel to the sum of all gray level values of the entire image indicates the proportion of information carried by that pixel. Actually, contractive maps along with the probabilities try to approximate the ratio of each gray level value to the sum of all gray level values. So, in the proposed algorithm, it is essential to store the information regarding the sum of gray level values. Also the corresponding contractive maps are nothing but the maps from a parent subimage to its child subimages [Figure 1]. The size of a parent subimage is double that of its child subimages. Hence, the map from parent subimage to its child is a contractive map.

We have already mentioned that the division of a parent subimage into its children depends on the performance of the approximation of the subimage by the Markov operators. Now if the complexity of the pixel values of the parent subimage is higher, then further division of the subimage is quite likely. On the other hand the image region where the complexity of the pixel values is low, no further division is required. In this case, if there is division, then we are losing in the sense of compression ratio. In order to avoid this problem we have introduced a simple classification scheme. The scheme classifies the child subimages into two groups according to the variability of the pixel values. If the variability is low *i.e.*, if the variance of the pixel values in the subimage is below a fixed value, called threshold, we call the subimage as smooth type. Otherwise we call it a rough type. Each pixel value in a smooth type subimage is replaced by the mean of all pixel values. The procedure can be looked upon as a condensation map. Note that under a condensation map, an image region is nothing but the copy of itself. Rough type subimages are approximated by the computed Markov operators. Hence, for each subimage we are to store either a probability value or the mean of the pixel values along with the information regarding the block type.

It may be noted that there will exist a level at which no further division of the image blocks is required. When the size of a child image block becomes 1×1 [*i.e.*, the child image block is a pixel], it can't be further subdivided. It may also be observed that, in such a case, there will be a contractive map from the parent image block to the child image block such that the error in estimating the pixel value of the child image block is zero.

The description of the encoding algorithm, the computation of bit rates and the decoding process are described in the following subsections.

3.1. Description of the Encoding Algorithm

Step 1: Compute the sum of gray level values of the given image.

Step 2: Partition the image (I) into its four quadrants. Each quadrant is called child subimage ($I_k, k = 1, 2, 3, 4$) of I .

Step 3: Compute

$$i) p_k = \frac{\text{sum of pixel values in } I_k}{\text{sum of pixel values in } I}; \quad \sum_{k=1}^4 p_k = 1.$$

$$ii) \text{Mean}(I_k) = m_k \text{ (say), } k = 1, 2, 3, 4.$$

$$iii) \text{Variance}(I_k) = v_k \text{ (say), } k = 1, 2, 3, 4.$$

Step 4: If $\text{Variance}(I_k) < Th$, [Th is a prefixed value]

then approximate I_k by replacing each pixel value of I_k by m_k ,

$$\text{else compute } I_k^1 = p_k \cdot w(I), \quad k = 1, 2, 3, 4. \quad (\text{A})$$

Here $w(I)$ is obtained by considering non overlapping windows of size 2×2 covering the entire image (I) and taking sum of four pixels within a window. thus, the size of $w(I)$ becomes equal to that of I_k , $k = 1, 2, 3, 4$.

Step 5: Compute I^1 by appending I_k^1 , $k = 1, 2, 3, 4$. (B)

Repeat the process (A) and (B) until $I^N = I^{N+1}$, where $I_k^N = p_k.w(I^{N-1})$ is the approximation of I_k . Note that I^N is an approximation of I by Theorem 6 [see Section 2]

Step 6: Check the difference between I_k and I_k^N .

If the difference is small (approximation is satisfactory)

then STOP,

else replace I by I_k^1 , $k = 1, 2, 3, 4$ and repeat steps 1, 2, 3, 4 and 5 until the size of I_k is equal to a prefixed value.

A schematic diagram of the encoding process is presented in Figure 2.

3.2. Computation of Bit Rate

The complete description of an image in terms of its code, with respect to its storage or transmission requirements, is to be evaluated to get the compression ratio or bit rate. The important overhead costs which are to be considered for the proposed encoding method of an image are : (i) the description of the image partition, (ii) the nature of the blocks, *i.e.*, class information of blocks and (iii) the quantized values of the numerical parameters.

First of all, the sum of the gray level values or the average gray level value of the given image is to be stored. If the given image is a 8 bit/pixel image then 8 bits are required to store this average gray level value. Next a complete description of the image partition needs to be stored. Note that we have used here a multiscaling type division. At any stage, a block is either partitioned into its child blocks or remains unaltered. The partitioning of a block into its child blocks depends on the performance of the estimator of the parent block. In particular, for a parent block, there may exist either one or two or three or all four children blocks. Therefore, there are, sixteen possible coding configurations for a block at any stage and 4 bits are required for storing this information. The image is partitioned in a quad-tree fashion. Whenever a image block is partitioned in to 4 sub-blocks, one needs to decide whether to partition a sub-block further or not. There could be 16 possible configurations available for this. If one needs to partition only one sub-block the information about which sub-block to be partitioned needs to be stored. So there are 4 such cases. Similarly, if one needs to partition 2 sub-blocks out of 4, then there will be 6 possibilities to indicate which two sub-blocks required further partitioning. Again if one needs to partition 3 sub-blocks out of 4, then there are 3 possibilities for the same. Lastly, there is only one possibility if one needs to partition all 4 sub-blocks. So, all together there are 16 possibilities available for each image block at level of partitioning. We call these possibilities as 16 configurations. Figure 3 shows the sixteen possible configurations of a block at any stage. Each block is again, either smooth type or rough type. Thus for storing block information, we need to store one more bit for each block. Now, we are left with two most important numerical parameters, the mean gray level value of a smooth type block and probability corresponding to a block. the mean values and the probabilities are quantized and stored in the codes.

3.3. Image Reconstruction from the Code

Image reconstruction is a process of decoding the obtained code. Here the decoding process is almost similar to the encoding process. The process starts with an arbitrary image with at least one pixel having nonzero pixel value. This starting image plays the role of a dummy image. In the decoding process no

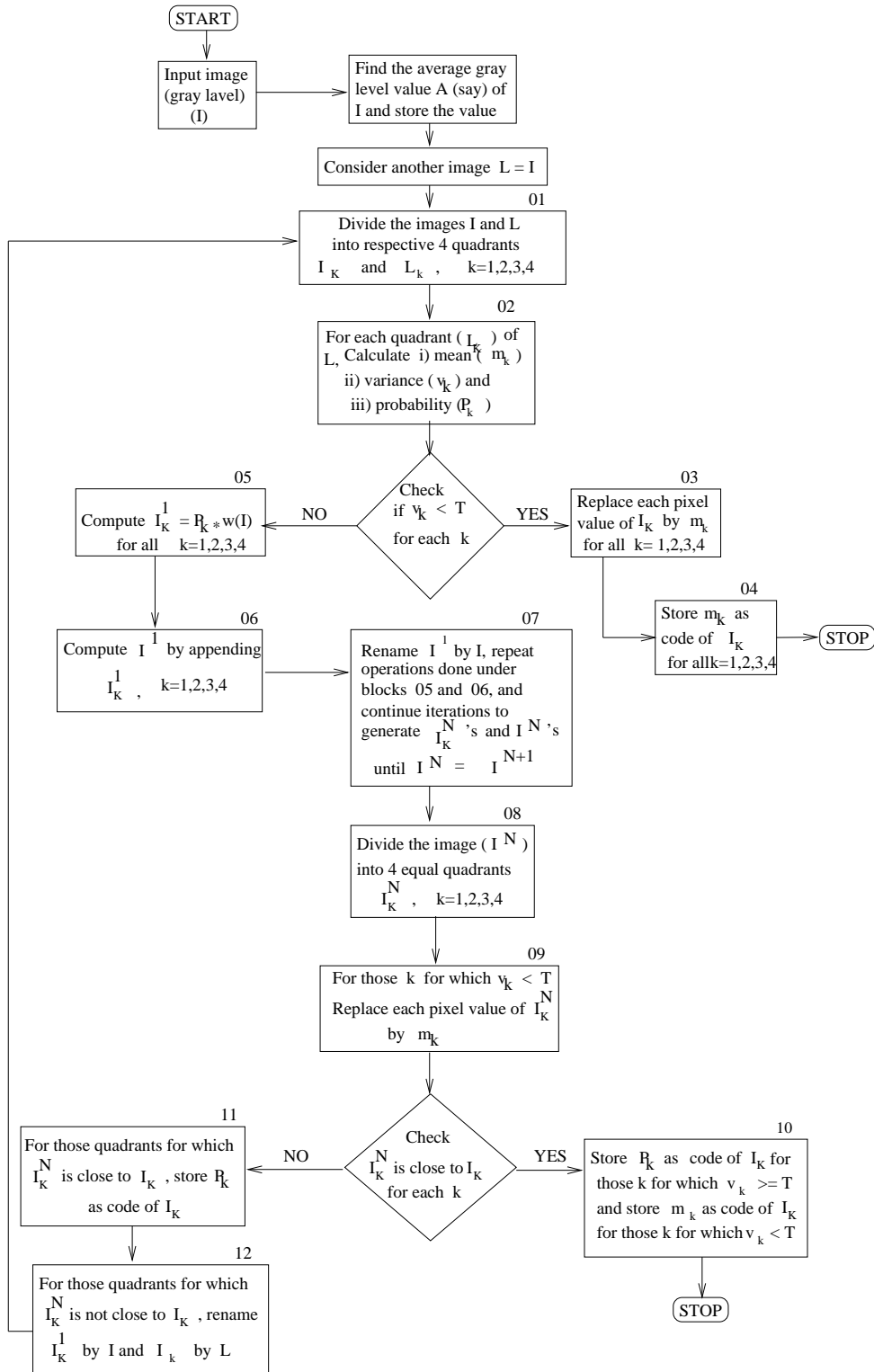


Figure 2. Schematic diagram of the proposed encoding algorithm

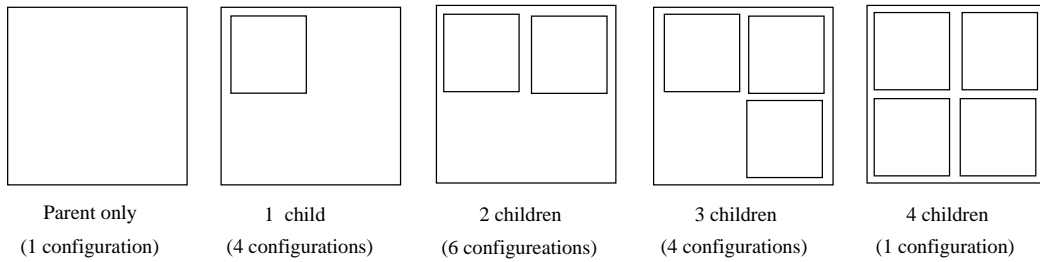


Figure 3. Possible configurations of a subimage

probabilities or contractive maps are computed. The starting image is partitioned following the partition rule stored in the codes. Using the block information, the mean gray level values or the stored probabilities and the maps are applied to the blocks of the starting image to get the stabilized value for each pixel of the starting image. These stabilized values are not the fixed points of the target image. The desired image [*i.e.*, the close approximation of the target image] is obtained by multiplying each stabilized value of the starting image by the ratio of sum of the gray level values of the original image, stored in the codes, to that of the starting image.

In the decoding, the number of iterations to get to the stabilized value of a pixel of the starting image is fixed, depending on the image size. Also, unlike the decoding of a PIFS code [12, 14, 15], no intermediate reconstruction of the image is possible in this decoding scheme. Note that, the starting image should be such that, there exist at least one pixel with nonzero pixel value. Otherwise, whatever may be the subdivision of the image, we will always obtain the image with every pixel value being zero.

A schematic diagram of the decoding process is presented in Figure 4.

4. Implementation and Results

The probabilistic approach for fractal image compression, discussed in Section 3 is tested with a wide range of gray scale images. We are presenting here the results obtained for some 256×256 , 8 bits/pixel images. First of all, let us discuss the results obtained by implementing the algorithm on the benchmark image “Lena” (256×256 , 8bits/pixel). The original image of “Lena” is shown in Figure 5. The image is subdivided into four 128×128 subimages, each of which is encoded separately. Partitioning of each subimage has been carried out up to the level of subimages of size 2×2 . The performance of the encoding methodology is also examined by considering the image partitioning up to the level of subimages of size 4×4 .

For encoding the “Lena” image, the results of partitioning up to 2×2 subimages and 4×4 subimages are reported here. Also, the encoding using partitioning up to 2×2 subimages has been performed for different threshold values. Note that, successful approximation of an image block at any stage will be judged by the selected threshold value. Actually, the threshold is set to evaluate the error criterion (here, the error criterion is RMSE). All the test results and some statistics of both the cases are given in Table 1. The test result is showing how compression ratio increases with decrease in the PSNR value for encoding up to 2×2 subimages. The time taken for encoding is also computed to show the relative fastness of the proposed fractal image compression scheme using IFS with probabilities. The programming language used here is “c” and all the programs have been executed on a 233 MHz Silicon graphics Workstation.

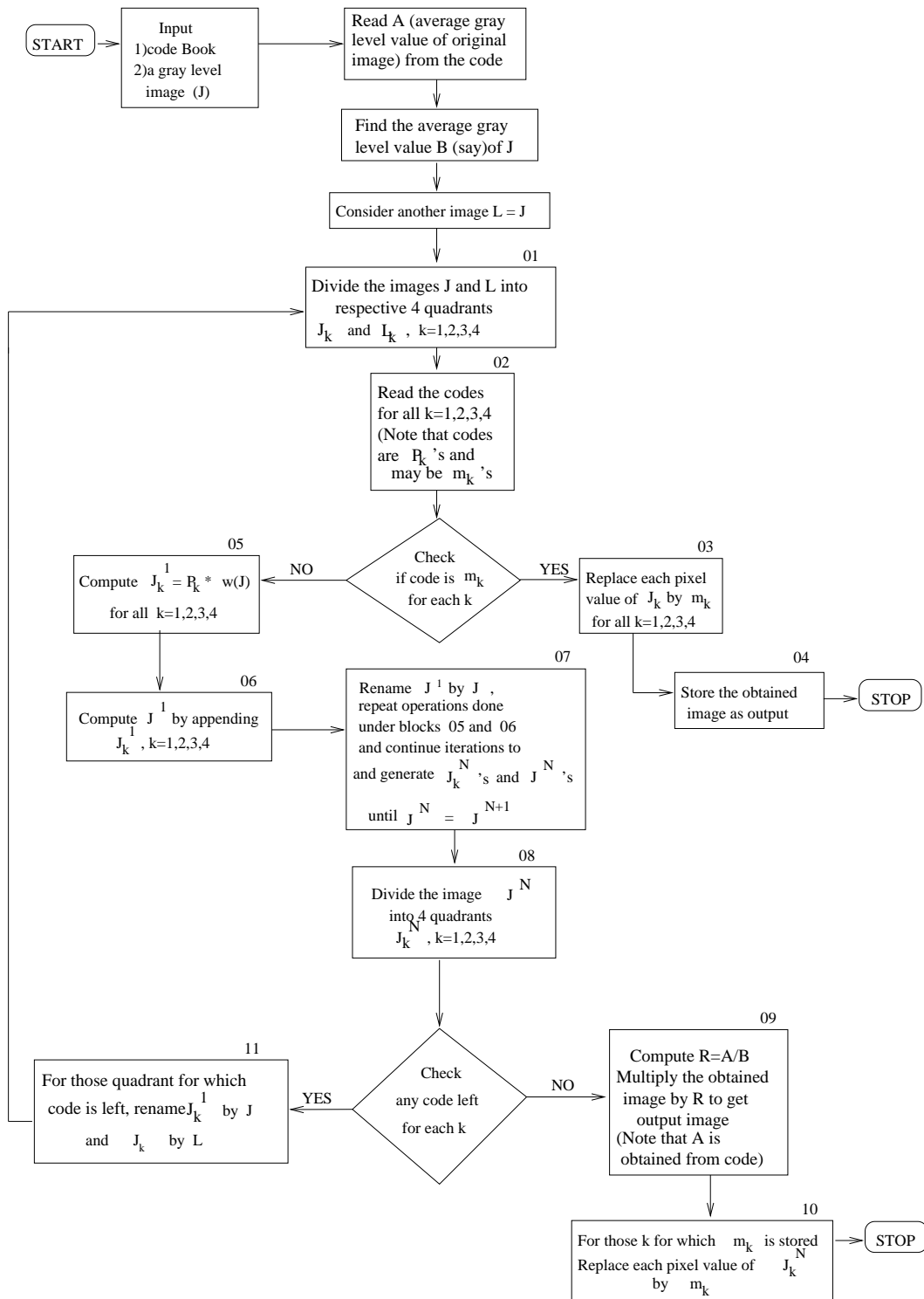


Figure 4. Schematic diagram of the proposed decoding algorithm



Figure 5. Original images of Lena, Girl, LFA, and Seagull (8 bpp)

Table 1. Test results for 256×256 , 8 bit/pixel “Lena” image using different threshold values

Lowest level subimage size	Compression Ratio	PSNR (in db)	bits/pixel (bpp)	Time elapsed (in Sec.)
4×4	20.1	22.43	0.40	8.34
2×2	3.20	30.49	2.50	23.39
2×2	5.08	28.23	1.58	16.40
2×2	8.81	27.11	0.90	10.93

The corresponding figures for the “Lena” image are shown in Figure 6. In Figure 6, the top left image shows the decoded image of “Lena” where the encoding is carried out up to the subimage of size 4×4 . The rest of the images are showing the decoded “Lena” for different threshold values when the encoding is processed up to the subimages of size 2×2 . The decoding process is carried out with a starting image which is “White” image having all the gray level values fixed at 255.



Figure 6. Decoded Lena (clockwise from top left corner): (i) 0.40 bpp, where subimage size is 4×4 ; (ii) 0.90 bpp, subimage size is 2×2 ; (iii) 1.58 bpp, where subimage size is 2×2 ; (iv) 2.50 bpp, where subimage size is 2×2

The algorithm is also tested for the “Girl” image, a “low flying aircraft (LFA)” image and a “Seagull” image. All these images are 8 bit/ pixel images of size 256×256 . The original images are shown in Figure 5. Experimental results of these images are shown in Table 2. Images in Figure 7 are the decoded images of “LFA”, “Girl” and “Seagull” respectively. For all the cases the starting image is considered as the “White” image having all the gray level values 255. Note that, images can also be reconstructed starting from any other image.

Table 2. Test results for several 256×256 , 8 bit/pixel images

Image	Compression Ratio	PSNR (in db)	bits/pixel (bpp)	Time elapsed (in Sec.)
Girl	10.41	27.75	0.76	9.68
LFA	4.46	24.23	1.79	15.46
Seagull	7.20	26.12	1.11	12.56



Figure 7. (Left to right): (i) decoded LFA, 1.79 bpp, where subimage size is 2×2 , (ii) decoded Girl, 0.76 bpp, where subimage size is 2×2 , and (iii) decoded Seagull, 1.11 bpp, where subimage size is 2×2

The proposed probabilistic approach for fractal image compression method is also compared, in terms of the computational time, with the GA based fractal image compression technique proposed by Mitra et al. [14]. The basic feature of GA based fractal image compression scheme is same as that of the scheme proposed by Jacquin [12] and is usually known as partitioned iterated function system (PIFS) based compression. A GA based compression technique tries to reduce the computational time for the implementation of the PIFS based compression technique. It has been found that almost 20 times reduction in the search space is achieved for the GA based compression technique in comparison to the usual PIFS based image compression technique. In the usual PIFS based image compression technique, an image is tiled with “range blocks”, and for each range block a mapping on a larger “domain block” is found such that the transformed domain approximate the range block. The search for an appropriate domain block and transformation for each range block is carried out by exhaustive search mechanism. In GA based technique, the search is performed using GAs. Note that, the present probabilistic approach is introduced with no search mechanism. The test results of both probabilistic approach encoding and the GA based encoding techniques are presented in Table 3.

Table 3. Results obtained by using probabilistic approach and the GA based technique for fractal image compression

Image	probabilistic approach			GA based technique		
	Compression Ratio	PSNR (in db)	Time elapsed (in Sec.)	Compression Ratio	PSNR (in db)	Time elapsed (in Sec.)
Lena	8.81	27.11	10.93	10.50	30.22	3141.47
Girl	10.41	27.75	9.68	11.37	30.74	2608.18
LFA	4.46	24.23	15.46	5.51	26.86	5838.17
Seagull	7.20	26.12	12.56	7.40	27.27	4240.02

It is very clear from the Table 3 that the proposed probabilistic approach for fractal image compression is very fast compared to the GA based fractal image compression technique. On the other hand, the compression ratio and the quality of the decoded image in terms of PSNR of the proposed method is not so impressive as compared to that of GA based technique. A few blocking effects, due to quantization, are visible in the decoded image of the present algorithm. But in respect of computational time, the proposed method is probably one of the fastest, if not the fastest.

5. Conclusions and Discussion

The proposed probabilistic approach for image compression is a direct outcome of the theory of IFS with probabilities. The algorithm is very fast in the sense of computational time. But at the same time, the technique is very much dependent on the partitioning scheme used. As we are using multiscaling type division to find the contractive maps and corresponding probabilities, it is assumed that, under a suitable transformation, there exists at least some similarity between image regions along this direction. But real life images need not necessarily possess similarity like this. To overcome this problem one has to divide the image up to subimages which are considerably small. Note that the algorithm will be a lossless one if one goes up to the pixel level division for every pixel. But, in that case, it would not be useful for compression as there will be hardly any compression.

The proposed method is also applicable in case of condensation maps included in the set of maps. A particular type of condensation map is used in the present case. Note that, due to this type of condensation map, each pixel value of an image region is replaced by the average of all pixel values of that region.

The present scheme is an attempt to find an image compression technique using IFS and probabilities. Some blocking effects are observed, though those are not very serious, in the final decoded image. It is true that some modifications have to be incorporated to make the algorithm more efficient to achieve a high compression ratio as well as quality decoded image. The information regarding multiscaling partition is taking a huge storage space causing reduction in the compression ratio. One way to increase compression could be to bypass this information. Towards the solution of this problem one can possibly make the multiscaling image subdivision mandatory up to a prespecified level. In particular, for the first few subdivision, one may consider all four subimages instead of going for option of selecting a few out of four depending on the image estimation obtained at that subdivision. This may reduce the burden of storing 4 bits for each subimage at each level. Another way of increasing compression could be done by adopting a lossless compression, *eg.* arithmetic or run length, scheme and applying it to the code obtained by the probabilistic approach.

The huge reduction in the computational time makes the present algorithm more attractive. The probabilistic approach technique for fractal image compression is at least 300 times faster than the GA based fractal image compression technique. The relative fastness of the proposed algorithm appeared to be its main feature. The method is found to be practically an online process. The fastness of the algorithm can be utilized to real time coding, for which reasonable compression may be delivered very rapidly. A U.S patent [1] on this work has already been obtained.

The present method of compressing an image is based on the fact that what proportion of pixel values is present in a small block of image compare to the total pixel value present in the entire image. It turns out to be a self similar in nature. More precisely, it is portioned self similar in nature. It is working well as no natural image shows exact self similar characteristics. However, K. Culik II et al.[17, 18, 19]

established a relation between Iterated Function Systems (IFS) and automata theory where self similar images could be generated using rational expression. These works opened up a new concept for image analysis. In the present discussion we have not tried to discuss the same.

References

- [1] Acharya, T., Bhattacharya, B. B., Kundu, M. K., Mitra, M., Murthy, C. A : Method of compressing an image, *U.S patent no. 6,738, 250 dt. May 18, 2004.*
- [2] Barnsley, M. F.: *Fractals Everywhere*, Academic Press, New York, 1988.
- [3] Barnsley, M. F., Hurd, L. P.: *Fractal Image Compression*, AK Peters Ltd., Massachusetts, 1993.
- [4] Dudbridge, F.: Linear time fractal coding schemes, in: *Fractal Image Encoding and Analysis, NATO ASI Series F, vol. 159* (Y. Fisher, Ed.), Springer Verlag, Berlin, 1998, 133–137.
- [5] Elton, J. H.: An ergodic theorem for iterated maps, *Ergodic theory and Dynamical Systems*, **7**, 1987, 481–488.
- [6] Fisher, Y.: *Fractal Image Compression: Theory and Application*, Springer Verlag, New York, 1995.
- [7] Fisher, Y.: *Fractal Image Encoding and Analysis. NATO ASI Series F, vol. 159*, Springer Verlag, Berlin, 1998.
- [8] Fisher, Y., Jacobs, E. W., Boss, R. D.: Fractal Image Compression Using Iterated Transforms, in: *Image and Text Compression* (J. A. Storer, Ed.), Kluwer Academic Publishers, 1992, 35–61.
- [9] Forte, B., Vrscay, E. R.: Solving the inverse problem for measures using iterated function systems : a new approach, *Adv. appl. Prob.*, **27**, 1995, 800–820.
- [10] Graf, S.: Barnsley's scheme for the fractal encoding of images, *Journal of Complexity*, **8**, 1992, 72–78.
- [11] Hutchinson, J.: Fractal and self similarity, *Indiana Univ. J. Math.*, **30**, 1981, 713–747.
- [12] Jacquin, A. E.: Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations, *IEEE Transactions on Image Processing*, **1**(1), 1992, 18–30.
- [13] Jacquin, A. E.: Fractal Image Coding : A Review, *Proceedings of the IEEE*, **81**(10), 1993, 1451–1465.
- [14] Mitra, S. K., Murthy, C. A., Kundu, M. K.: Technique for fractal image compression using Genetic Algorithm, *IEEE Transactions on Image Processing*, **7**(4), 1998, 586–593.
- [15] Mitra, S. K., Murthy, C. A., Kundu, M. K.: Partitioned iterative function system: A new tool for digital imaging, *IETE Journal of Research*, **46**(5), 2000, 363–370.
- [16] Thomas, L., Deravi, F.: Region-Based Fractal Image Compression using Heuristic Search, *IEEE Transactions on Image Processing*, **4**(6), 1995, 832–838.
- [17] K. Culik II and S. Dube: Rational and affine expressions for image description, *Discrete Applied Mathematics*, **41**, 1993, 85–120.
- [18] K. Culik II and J. Kari: Image Compression Using Weighted Finite Automata, *Computer Graphics*, **17**(3), 1993, 305–313.
- [19] K. Culik II and J. Kari: Image-data Compression Using Edge-Optimizing Algorithm for WFA Inference, *Journal of Information Processing and Management*, **30**(6), 1994, 829–838.