ISSN 2347-1921

Fractal Image Compression Using Modified Operator(IFS)

Rashad A. Al-Jaw[1], Baligh M. Al-Helali[2], Adil M. Ahmed[3]
[1,2]Department of Mathematics and Computer Science, Faculty of Science, Ibb University, Yemen
[3]Department of Mathematics, Faculty of Ibn Alhaitham for education, Baghdad University, Iraq
[1]algofi@yemen.net.ye, [2]baleegh.helali@gmail.com, [3]adilm.ahmed@yahoo.com

ABSTRACT

Image data Compression based on fractal theory is fundamentally different from conventional compression methods, its idea is to generate a contraction operator whose fixed point approximates the original image in a complete metric space of images. The specification of such operator can be stored as the fractal code for the original image. The contraction mapping principle implies that the iteration of the stored operator starting from arbitrary initial image will recover its fixed point which is an approximation for the original image. This contraction mapping is usually constructed using the partitioned IFS(PIFS)technique which relies on the assertion that parts of the image resemble other parts of the same image. It then, finds the fractal code for each part by searching for another larger similar part. This high costly search makes fractal image compression difficult to be implemented in practice, even it has the advantages of a high compression ratio, a low loss ratio, and the resolution independence of the compression rateo.

In this paper, we investigate fractal image compression (FIC) using Iterated Function Systems (IFS). After reviewing the standard scheme, we state a mathematical formulation for the practical aspect. We then propose a modified IFS that relies on the fact that, there are very constant parts in certain images. From the view point of mathematics, we present the modified operator, proving its properties that make it not only a fractal operator but also more effective than the standard one. The experimental results are presented and the performance of the proposed algorithm is discussed.

Keywords:

Fractal; Image comporation; Iterated Function System (IFS)

ISSN 2347-1921

1. INTRODUCTION

The mathematics behind fractals began to take shape in the 17th century when mathematician and philosopher Leibniz considered recursive self-similarity (although he made the mistake of thinking that only the straight line was self-similar in this sense) [6]. Fractal image compression has been studied from different perspectives [11] such as: Iterated Function Systems, Self vector quantization [9], Self-quantized wavelet sub trees [13], and Convolution transform coding [14]. This motivated Barnsley to search for an image compression system by modeling images as attractors of IFSs [3]. Barnsely suggested that storing the fractal like image as a collection of transformations reduces the required memory of storing the image as a collection of pixels. Iterated functions in the complex plane were investigated in the late 19th and early 20th centuries by Henri Poincar, Felix Klein, Pierre Fatou and Gaston Julia. However, without the aid of modern computer graphics, they lacked the means to visualize the beauty of many of the objects that they had discovered [8]. In the 1960s, Benot Mandelbrot started investigating self-similarity in papers such as How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension, which built on earlier work by Lewis Fry Richardson. Finally, in 1975 Mandelbrot coined the word fractal to denote an object whose Hausdorff-Besicovitch dimension is greater than its topological dimension [8]. Fractal image compression (FIC) was introduced by Barnsley and Sloan [1]. They introduce in another work a better way to compress images [2], and after that, (FIC) has been widely studied by many scientists. FIC is based on the idea that any image contains self-similarities, that is, it consists of small parts similar to itself or to some big part in it. So in FIC iterated function systems are used for modeling. Jacquin [12] presented a more exible method of FIC than Barnsley's, which is based on recurrent iterated function systems (RIFSs) introduced first by him. RIFSs which have been used in image compression schemes consist of transformations which have a constant vertical contraction factor. Fisher [10] improved the partition of Jacquin. A hexagonal structure called the Spiral Architecture (SA) [4] was proposed by Sheridan in 1996. Bouboulis etc. [7] introduced an image compression scheme using fractal interpolation surfaces which are attractors of some RIFSs. Kramm presented a quite fast algorithm [5], manages to merge low-scale redundancy from multiple images.

2. FRACTALS ENCODING USING ITERATED FUNCTIONS SYSTIMS

2.1 Iterated Function Systems

The concept of iterated function systems (IFS) was first developed by Hutchinson [25], then independently discovered by Barnsley and Demko [24] who gave them their name.

Definition 2.1 Let $w_i : X \to X$ be a mapping on X for $i = 1,2,3,\ldots,N$, then we will referto $\{X; w1, w2, \ldots, wN\}$ as an iterated function system or IFS.

If the functions $w1, w2, \ldots, wN$ are contractions then the IFS that compound of them issaid to be contractive. This property is useful in generating fractals by such systems andfor this purpose the concept of IFS is extended from X to the space of fractals $H(X)$ whichdefined as follows.

Definition 2.2 (Hausdorff metric Space) [26] Let (X, d) be a metric space, the Hausdorff space $H(X)$ denotes the set of all non-empty compact subsets of X.

Let (X, d) be a metric space, and $H(X)$ be the associated Hausdorff_ space. Then thedistance from a point $x \in X$ to B $\in H(X)$ is defined by

$D_B(x) = \min\{d(x,b) : b \in B\}$

Moreover, the distance from A to B is defined by

$$D_B(A) = \max\{D_B(a) : a \in A\}$$

$D_B(x) = \min\{d(x,b) : b \in B\}$

Moreover, the distance from A to B is defined by

$$D_B(A) = \max\{D_B(a) : a \in A\}$$

These definitions make sense because D_B (x) is a continuous function of $x \in$A [23] and A is compact being in $H(X)$, so there must exist a point $\hat{a} \in$ A such that,$D_B(\hat{a}) \geq D_B(a) \forall a \in A$.

It is necessary to notice that, it is possible to find two points in the Hausdorff space$(A, B \in H(X))$ such that $D_B(A) \neq D_A(B)$. Hence the distance d(A,B) = D_B(A) does not constitute a metric. The next theorem defines the Hausdorff distance that constitute a metric on the Hausdorff space.

Theorem 2.1 Let (X, d) be a metric space and $H(X)$ denote the nonempty compact sub sets of X. Define

$$h(A,B) = \max\{D_B(A), D_A(B)\} \forall A, B \in H(X).$$

Then $(H(X), h)$ is a metric space.

Proof: See [22]

There are many properties that induced from X to $H(X)$ as refereed in [23, 22]. Forexample,if (X, d) is complete then $(H(X), h)$ is also complete, and the contraction of $w : X \to X$ isinherited by $w : H(X) \to H(X)$.

These properties enables the extension of IFS from X to $H(X)$, and ensure that each$w_i$is continuous and maps $H(X)$ into itself, which allows the Hutchinson operator to be defined as follows.

Definition 2.3 The Hutchinson operator of a finite set of maps $\{W_i\}_{i=1}^{n}$on a set A is written as:

$$W(A) = \cup_{i=1}^{N} w_i (A) \quad (1)$$

The Hutchinson operator is a convenient tool for later proofs, and the inheritance of completeness is important because it used to proof the existence of many fractals. The following theorem provides a general condition for the completeness of $(H(X), h)$, it also characterizes the limits of Cauchy sequences in $(H(X), h)$.

Theorem 2.2 Let (X, d) be a complete metric space. Then $(H(X), h)$ is a complete metricspace. Moreover, if $\{ A_n \in H(X)\}_{n=1}^{\infty}$ is a Cauchy sequence then,

$$A := \lim_{n\to\infty} A_n ,$$

can be characterized as $A = \{x \in X |\ \text{there is a Cauchy sequence that convergesto}\ x\}$.

proof See [23, 22].

The unique fixed point $A_W \in H(X)$ of$W = \cup_{i=1}^{N}$ wi is called a fractal in many references [20].

2.2 Fractal Encoding using IFS

2.2.1 Classical Fractals Compression

We can notice that, many classical fractals may be represented as binary (black-and-white) images on computers. Since many such fractals are generated by IFS as seen previously, then there are many images that can be generated by IFS. The question now will be as, Which better to represent such images on computer memory, IFS code (parameters), or the traditional pixel array. The answer may be induced by the following example [28, 27]:

Figure 1: A simple fractal tree.

Example 2.3 The tree-like image in Figure 1 is generated by IFS of four transforms, listed in Table 2. This IFS can be stored in a file of size 176 bytes, whereas the pixel array representation of the original image requires 264,000 bytes, achieving 1,500 compression ratio.

Transform	A	B	C	D	E	F
W1	0.53	-0.08	0.08	0.53	-0.88	33.44
W2	-0.31	-0.42	-0.44	0.33	-15.19	19.43
W3	-0.25	-0.05	-0.07	0.29	1.48	11.73
W4	0.29	0.54	-0.04	0.29	18.74	9.87

Table 2: The parameters of the four affine transforms resembling the IFS that generates the tree in Figure 1

In this example, the generating IFS code is known, however, we do not know the IFS generator for all binary images. Moreover we do not know if such IFS exists or not. So the problem will be: For a given image A, how to find a contractive IFS whose attractor is this image. This problem is known as the inverse problem for iterated function systems [21, 19].

To address this problem mathematically, we have to state a suitable mathematical framework. Let X be a compact subset of R^2 then it is possible to consider $H(X)$ as a collection of all binary (black-and-white) images, where a subset of the plane is represented by an image that is black at the points of the subset and white elsewhere. As mentioned previously, many classical fractals are elements in Hausdorff space $H(X)$ which denotes the set of all non-empty compact subsets of X.

2.2.2 Collage Theorem On $(H(X), h)$

Since the number of points in fractal sets is infinite and complicatedly organized, it is difficult to specify exactly the generator IFS. From the practical point of view, it will be acceptable for the required IFS to be chosen such that its attractor is close to a given image for a pre-defined tolerance.

The collage theorem is very useful to simplify the inverse problem for fractal images [23], it has been addressed by many researchers as well [19].

Theorem 2.4 Let $\{X; w1, w2, \ldots, w_N\}$ be a hyperbolic IFS with contractivity factor s, and W be the associated Hutchinson map as defined in Eq.(1) then

$$h(A, A_W) < \frac{h(A, W(A))}{1-s}, \forall A \in H(X) \tag{2}$$

where A_W is the fixed point of W.

Hence if $h(A, W(A)) < \epsilon$ then

$$h(A, A_W) < \frac{\epsilon}{1-s}.$$

Proof: see [23]

The theorem can be used as following. Given a fractal image A, find a set of contractive mappings that maps A into smaller copies of itself such that the union of the smaller copies is close as ϵ to the target image. The determined contractions are the IFS codes with corresponding Hutchinson operator W.

The theorem states that, the attractor AW of the determined IFS W approximates the target image A $(i.e., h(A, AW) < \frac{\epsilon}{1-s}$. It also implies that, the more accurately the IFS maps the image to itself, the more accurately the IFS approximates the image. The consequence of collage is the following useful corollary.

Corollary 2.5 Let $A \in H(X)$, Given ϵ> 0,there exist a hyperbolic IFS $\{X; w1, w2, \ldots, w_N\}$ with attractor AW satisfying

$$h(A, A(W)) < \epsilon.$$

Proof: see [17, 18].

The attractor of the IFS is not necessarily equal to the original image, it can be a close approximation to it, so the fractal compression is lossy.

Clearly, the self-affine images are more accurately represented by affine IFS than other images that require an approximate solution.

2.3 Fractal Image Compression using Local Iterated Function Systems

The classic IFS methods represent a target image as a union of small copies of itself. This representation is suitable for the entirely self-similar images, but most real-world images do not have this type of similarity. Instead, they may have another type of similarity where sub portions of the target image may be similar to other subregions in the same image. These images are said to be locally self-similar, and this was the idea of block-based fractal-based image compression scheme which developed by Jacquin, in the late 1980s.

2.3.1 Local(Partitioned) Iterated Function System

Most natural images are not seemed to have the self-similarity which found in classical fractals, however they have a self-similarity of another type that is small region of an image is seen to be similar to a large region of the same image which called quasi-self-similarity as the remarked regions in Figure 2.

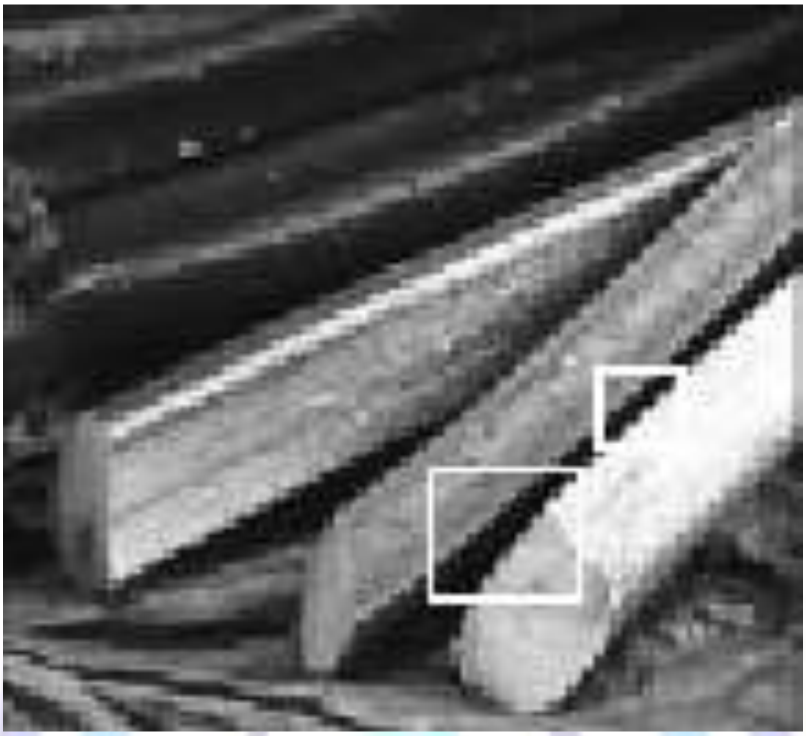

Figure 2: Self-similar portions in a neutral image

This type of self-similarity can be seen in most images that found in the real-world, such as, images of faces, trees, mountains, clouds, houses, etc. This leads to the idea of representing images by transformed parts of themselves.

For this purpose, a generalization of the iterated function system concept is used, this extension is based on that the maps $\mathrm{w1}, \ldots, \mathrm{wN}$, are applied to local domains of the image instead of the whole image. This variation is called local iterated function systems (LIFS) [3] or partitioned iterated function system (PIFS) [10, p. 48] and defined as following.

Definition 2.4 Let X be a complete metric space, and let $D_i \subset X$ for $i = 1, 2, \ldots, N. A$ partitioned (local) iterated function system is a collection of contractive maps $w_i : Di \rightarrow X$, for $i = 1, 2, \ldots, N$.

It is clear from the definition that, the specification of each w_i cannot be done only using an contractive mapping, it also needs the domain of this mapping.

2.3.2 Image Encoding using Local IFS

IFS-type methods sought to express a target set or image as a union of shrunken copies of itself. However, as before most real-world objects are rarely so entirely self-similar. Instead, self-similarity may be exhibited only locally, in the sense that subregions of an image may be self-similar. In the late 1980s, Jacquin developed a block-based fractal image compression scheme that exploits local self-similarities within images [12]. This fractal-based scheme is based on exploiting the inherent local self-similarities in the spatial domain of images. In fact, most real-world images exhibit some degree of local self-similarity which can be exploited by using fractal-based image compression methods. To exploit the local self-similarities within sub-regions of images, the image is subdivided into a pair of simple and uniform partitions of the image: A domain partition of larger sub-blocks, also known as parent sub-blocks and a range partition of smaller sub-blocks, also known as child sub-blocks. A parent sub-block is mapped into its corresponding child sub-block using a geometric mapping, followed by a simple affine transformation, known as the gray-level map.

2.4 Generalized Fractal Image Encoding

Consider a complete metric space of images (Y, dY), and let $y_T \in Y$ be the fixed point of acontractive transformation T. The image $y_T$can be closely approximated by a convergingalgorithm that based on the contraction mapping principle. This algorithm suggests to iterate the transformation T starting from any initial image $y0$. This iteration processproduces the images sequence $\{T^{(n)}(y0)\}_{n=0}^{\infty}$that converges to the attractor of T which is its fixed point y_T. In practice, only few iterations are enough to approximate $y_T$within areasonably small error tolerance.

The image $y_T$is also called a fractal, because as a fixed point of a contractive transformation it often shows some fractal properties such as self-tiling and symmetry. Hence in fractal image compression, given a target image $\hat{y} \in Y$, the goal is to approximate $\hat{y}$ by a fractal y_T. For this purpose, the fractal encoding aims to construct an appropriate contractive transformation T whose attractor y_T closely resembles a given target image $\hat{y}$.

The problem of finding such operator T is called the fractal inverse problem or the fractal image coding problem, and can be stated mathematically as following:

Given a target image $\hat{y} \in Y$, construct a transformation T such that T is contractive on (Y, dY) and its fixed point, y_T, closely approximates $\hat{y}$.

The collage theorem is helpful in solving this question and can be stated for images as following.

Theorem 2.6 (Images Collage Theorem) Let $\hat{y}$ be a target image in a complete metricspace of images (Y, dY) and T is a contractive transformation defined on Y with contractivityfactor s, such that

$dY(\hat{y}, T(\hat{y})) <$, for some $\epsilon > 0$ (3)

then

$d_Y(\hat{y}, y_T) < \frac{\epsilon}{1-s}$ (4)

where y_T is the fixed point of T.

Proof See [16].

In view of this theorem, if a contractive transformation T maps the target image $\hat{y}$ close to itself, then the target image $\hat{y}$ will be closely approximated by the fixed point y_T of T. Consequently, the fractal image encoding problem can be reformulated as following:

Given a target image $\hat{y}$, find a contractive transformation T that maps $\hat{y}$ closest to itself.

Solving this question can be reduced to solving the following minimization problem for the parameters of T:

Minimize : $\mathrm{dY}(\hat{y}, \mathrm{T}(\hat{y}))$ subject to Tisconntractive. (5)

2.5 A Mathematical Formulation For IFS-based)

From the previous discussion we can say that, the IFS-based fractal image compression has the following mathematical issues:

1. The identification of suitable metric spaces (Y, d_y) that can be used to represent"images" supported on a region $\boldsymbol{X}$.

2. The construction of suitable fractal transform operators $T: Y \rightarrow Y$ over these spaces.It is also required to determine the contractivty conditions of T in (Y, d_y).

3. The formulation of fractal inverse problem in (Y, d_y), and a solution for such problemwhere a "target" element $f \in Y$ is approximated by a fixed point $f_T$of a contractivefractal transform T.

Now, we will follow these steps to provide a mathematical framework for a practicalfractal image coding scheme. We will focus on grey scale images that are more generalizedthan black-white images and can be studied mathematically better than color images thatare extensions of the grey-scale images. Let us identify a suitable metric spaces to represent"images" and present its mathematical properties to apply IFS-based FIC.

Let X be a screen (finite subset of $\mathbb{R}^2$), and d be a metric on$\mathrm{X}$, such that (X, d) be acomplete metric space, each point $\boldsymbol{x} \in \boldsymbol{X}$ is called a pixel. Then, from [15] $(\boldsymbol{B}(\boldsymbol{X}), \|.\|_\infty)$ isa Banach space where for each $\boldsymbol{f}: \boldsymbol{X} \rightarrow \mathbb{R}$, define

$$\|\boldsymbol{f}\|_\infty = \mathbf{max}\{|\boldsymbol{f}(\boldsymbol{x})| : \boldsymbol{x} \in \boldsymbol{X}\}$$

and

$$\boldsymbol{B}(\boldsymbol{X}) = \{\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{R}: \|\boldsymbol{f}\|_\infty < \infty\}$$

That means that, $(\boldsymbol{B}(\boldsymbol{X}), \|.\|_\infty)$ is a complete metric space, and if $\boldsymbol{T}$ be a contractionmapping on B(X) with contraction factor $\boldsymbol{\alpha}$, then $\boldsymbol{T}$ has a unique fixed point $\boldsymbol{f_T} \in \boldsymbol{B}(\boldsymbol{X})$.

For simplicity we will write $\mathrm{Con}(\mathrm{B}(\mathrm{X}))$ to refer to the set of contraction mappings on $(\mathrm{B}(\mathrm{X}), \|.\|_\infty)$. Now, to encode a given image $\mathrm{f}^* \in \mathrm{B}(\mathrm{X})$, it is required to find a contractive mapping $\mathrm{T} \in \mathrm{Con}(\mathrm{B}(\mathrm{X}))$ whose fixed point $\mathrm{f_T}$ is f^*, but in practical aspect it is enough for $\boldsymbol{f_T}$ to beclose to $\boldsymbol{f}^*$, so our aim is to answer the question: For a given image $\boldsymbol{f}^* \in \boldsymbol{B}(\boldsymbol{X})$ and $\boldsymbol{\epsilon} > 0$,can we find a non constant $\boldsymbol{T} \in \boldsymbol{Con}(\boldsymbol{B}(\boldsymbol{X}))$ such that $\|\boldsymbol{f}^* - \boldsymbol{f_T}\|_\infty < \epsilon$? Let us define the following subset of $\boldsymbol{B}(\boldsymbol{X})$

$$\boldsymbol{\gamma}(\boldsymbol{f}^*, \boldsymbol{\epsilon}) = \{\boldsymbol{f} \in \boldsymbol{B}(\boldsymbol{X}): \|\boldsymbol{f}^* - \boldsymbol{f_T}\|_\infty \leq \boldsymbol{\epsilon}\},$$

which forms a closed ball in $\boldsymbol{B}(\boldsymbol{X})$.

Thus we are looking for a contractive mapping $\boldsymbol{T}$whose fixed point $\boldsymbol{f_T}$be in $\boldsymbol{\gamma}(\boldsymbol{f}^*, \boldsymbol{\epsilon})$, the basin of $\boldsymbol{f}^*$. The ability of constructing such an $\boldsymbol{T}$ is uncertain at this stage. Moreover,it is uncertain if it even exists or not. So it is better to begin with another easier question:

Given $\boldsymbol{f}^* \in \boldsymbol{B}(\boldsymbol{X}), \boldsymbol{T} \in \boldsymbol{Con}(\boldsymbol{B}(\boldsymbol{X}))$, how close is $\boldsymbol{f}^*$ to $\boldsymbol{f_T}$? The following theorem is helpful to lend an answer:

Theorem 2.7 (Local Banach Contraction Principle) Given $f^* \in B(X)$ and $\epsilon > 0$, let T be a contraction mapping with contraction factor α. If

$$\frac{1}{1-\alpha}\|f^* - T(f^*)\|_\infty < \epsilon,$$

then T has a unique fixed point$f_T \in \gamma(f^*, \epsilon)$.

Proof: See [20].

This theorem enables us to formulate the following question. Given $f^* \in B(X)$ and $\epsilon > 0$,

can we find a non constant $T \in Con(B(X)$ such that $\frac{1}{1-\alpha}\|f^* - T(f^*)\|_\infty < \epsilon$?

When T is contraction we call it a fractal operator and if it satisfies the previous condition for some image f^*, then it called its fractal encoder.

After establishing a complete metric space of grey-scale images, it is possible now to construct a fractal transform that will be used later in encoding images. Let $R = \{R_i: i \in I\}$ be a partition for X, i.e., $X = \cup_{i\in I} R_i$ and $R_i \cap R_j = \emptyset, i \neq j$, and $D = \{D_j: j \in J\}$ bea cover for X, i.e.,$X = \cup_{j\in J} D_j$, where I, J are finite subsets of N.

The notations, R, D, R_i, D_i are called range pool, domain pool, range block, and domain block respectively, where R_i are a non overlapping sub regions of X and smaller than D_i.

For each $(i; j) \in I \times J$, let $\{w_{ij}\}$ be a contraction mapping (X, d), defined such that: $w_{ij}(D_j) = R_j \forall\, (i, j) \in I \times J$. For simplicity, we configure the $R_i$and $D_j$blocks to be squarepixel blocks, also the length and the width of the domain blocks are twice the length andthe width of the range blocks, as illustrated in Figure 4, i.e. the area of a domain block isfour times the area of the range block.

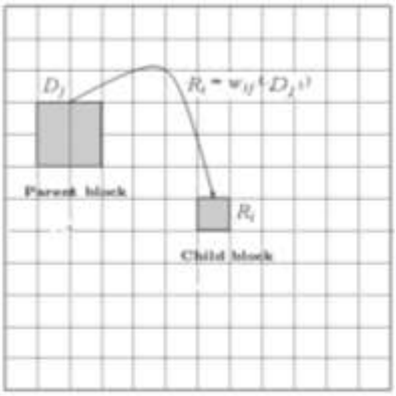

Figure 4: Spatial contraction $w_{ij}$of a domain block $D_j$to a range block R_i.

The contraction of these geometric mappings can be easily done on the continuousspaces.However, the shrinking of a parent block to a child block is not straight-forward ina discrete pixel space. It can be done by replacing neighbouring pixels in the parent blockby a single pixel and replacing the gray-level values by their average value.

Let $\{\tau_k\}_{k\in K}$, be a symmetric group on D_j, i.e., $\tau_k: D_j \to D$ and $\tau_k(D_j) = D_j$. Such isometry can be composited with a non-rotating affine transformation $w_{ij}$to definea geometric mapping $w_{ijk}$from $D_j$to $R_i$as:$w_{ijk} = w_{ij}\tau_k$ We will consider only the eightessential isometries of the K possible isometries [16], that act on $D_j$as shown for a specialcase in Figure 5. If $w_{ijk}(D_j) = R_j$, then the block D_j is called a parent block for the childblock R_j.

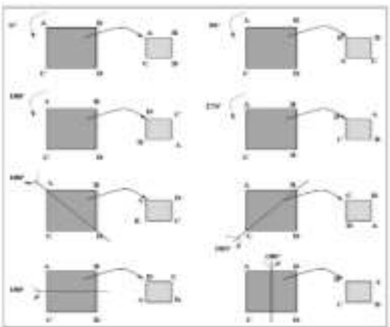

Figure 5: There are 8 geometric maps that transform a square parent block into a smaller square child block.

Now, we define the gray-level maps, that will be applied to these transformed parentsub-blocks. Let $\boldsymbol{f} \in \boldsymbol{B}(\boldsymbol{X})$ be a given image, $\boldsymbol{f}|_{\boldsymbol{R}_i}$is the sub image which restricted on $\boldsymbol{R}_i$,and can be denoted as $\boldsymbol{f}_i$such that for all $\boldsymbol{x} \in \boldsymbol{X}$:

$$\boldsymbol{f}_i(\boldsymbol{x}) = \begin{cases} \boldsymbol{f}(\boldsymbol{x}) & \boldsymbol{x} \in \boldsymbol{R}_i \\ \boldsymbol{0} & , \boldsymbol{x} \notin \boldsymbol{R}_i \end{cases} \qquad (6)$$

It is clear that $\boldsymbol{f}(\boldsymbol{x}) = \sum_{i \in I} \boldsymbol{f}_i(\boldsymbol{x})$for all $\boldsymbol{x} \in \boldsymbol{X}$ and $\boldsymbol{f}_i \in \boldsymbol{B}(\boldsymbol{X})$. Moreover, if we define $\boldsymbol{f}_{ijk}(\boldsymbol{x})$ to be the average of the values in the set $\boldsymbol{w}_{ijk}^{-1}(\boldsymbol{x})$for all $\boldsymbol{x} \in \boldsymbol{X}$, i.e.

$$\boldsymbol{f}_{ijk}(\boldsymbol{x}) = \boldsymbol{avr}\{\boldsymbol{f}(\boldsymbol{y}) : \boldsymbol{y} \in \boldsymbol{w}_{ijk}^{-1}(\boldsymbol{x})\},$$

then $\boldsymbol{f}_{ijk}(\boldsymbol{x}) \in \boldsymbol{B}(\boldsymbol{X})$. Once the pixel sizes of the parent blocks $\boldsymbol{D}_j$are reduced to be similar to that of the child blocks $\boldsymbol{R}_i$, we can define the following affine transformations:

$$\boldsymbol{T}_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(\boldsymbol{f})(\boldsymbol{x}) = \begin{cases} \boldsymbol{\alpha}_{ijk}\boldsymbol{f}_{ijk}(\boldsymbol{x}) + \boldsymbol{\beta}_{ijk} & \boldsymbol{x} \in \boldsymbol{R}_i \\ \boldsymbol{0} & , \boldsymbol{x} \notin \boldsymbol{R}_i \end{cases} \qquad (7)$$

$$, (\boldsymbol{\alpha}_{ijk}, \boldsymbol{\beta}_{ijk}) \in \boldsymbol{R}^2$$

In other words, $\boldsymbol{T}_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(\boldsymbol{f})(\boldsymbol{x})$ (also called the fractal component), represents a modified value of the grey level of f at the ithpreimage of x (if it exists). And because $\boldsymbol{B}(\boldsymbol{X})$ isa Banach space then $\boldsymbol{T}_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(\boldsymbol{f}) \in \boldsymbol{B}(\boldsymbol{x})$ for all $\boldsymbol{f} \in \boldsymbol{B}(\boldsymbol{X})$. Next proposition states that,the images under $\boldsymbol{T}_{ijk}^{(\alpha_{ijk},\beta_{ijk})}$, can only get so far from each other, depending on how closethey were before and the value of $\boldsymbol{\alpha}_{ijk,}$.

Proposition 2.8 The operator $T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}$in Equation (7) is lipschitz, with lipschitz constant$|\alpha_{ijk}|$, i.e.

$$\left\| T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(f) - T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(g) \right\|_\infty \leq |\alpha_{ijk}| \|f - g\|_\infty \qquad (8)$$

$\forall f, g \in B(X)$

Proof:

Let $f, g \in B(X)$, then$(f - g) \in B(X)$, and $\alpha_{ijk}(f - g) \in B(X)$, where $\alpha_{ijk} \in \mathbb{R}$,

because $B(X)$ is a vector space. And we have that, $(\forall x)(x \in R_i)(R_i \subseteq X)$,

$$|(f - g)_{ijk}(x)| \leq \|f - g\|_\infty \forall x \in X. \qquad (9)$$

Now, from the definition of $T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}$ in Eq.(7),

$$T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(f)(x) - T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(g)(x) = (\alpha_{ijk} f_{ijk}(x) + \beta_{ijk}) - (\alpha_{ijk} g_{ijk}(x) + \beta_{ijk})$$

$$= \alpha_{ijk}\left(f_{ijk}(x) - g_{ijk}(x)\right) = \alpha_{ijk}\left(\frac{\sum_{y \in w_{ijk}^{-1}(x)} f(y)}{\sum_{y \in w_{ijk}^{-1}(x)} 1} - \frac{\sum_{y \in w_{ijk}^{-1}(x)} g(y)}{\sum_{y \in w_{ijk}^{-1}(x)} 1}\right)$$

$$= \alpha_{ijk}(f - g)_{ijk}(y), \text{ where } y \in w_{ijk}^{-1}(x)$$

Using Eq.(9), we deduce that

$$\left| T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(f)(x) - T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(g)(x) \right| \leq |\alpha_{ijk}| \|f - g\|_\infty \forall x \in X. \qquad (10)$$

Therefore,

$$\left\| T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(f) - T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(g) \right\|_\infty \leq |\alpha_{ijk}| \|f - g\|_\infty, \qquad \forall f, g \in B(X) \blacksquare$$

The next task is to define the operator T which combines the distinct fractal components $T_{ijk}^{(\alpha_{ijk},\beta_{ijk})}(f)(x)$.

Let $i \in I$, and $j(i) \in J, k(i) \in K$be an associated domain and isometry indices.

$$T(f)(x) = T_{ij(i)k(i)}^{(\alpha_{ij(i)k(i)},\beta_{ij(i)k(i)})}(f)(x) : x \in R_i, \qquad (11)$$

It is easy to verify that

$$\|T(f)\|_\infty = \max\left\{\left\|T_{ij(i)k(i)}^{\left(\alpha_{ij(i)k(i)},\beta_{ij(i)k(i)}\right)}(f)\right\|_\infty, i \in I\right\} \quad (12)$$

From this construction, the transform T will define an operator $T: B(x) \to B(x)$ that associates to each image function $f \in B(X)$ the image function Tf, but the next theorem will define it in alternative way proving its important properties.

Theorem 2.9 Let T be defined on $B(X)$as

$$T(f)(x) = T_{ijk}^{\left(\alpha_{ij(i)k(i)},\beta_{ij(i)k(i)}\right)}(f)(x), \forall x \in X, \quad (13)$$

for all $f \in B(X)$, then

$$T: B(x) \to B(x)$$

Moreover,

$$\|T(f) - T(g)\|_\infty \leq \alpha\|f - g\|_\infty, \quad \forall f, g \in B(X)$$

where $\alpha = \max_{i \in I}|\alpha_{ij(i)k(i)}|$.

Proof:

Let $f \in B(X)$, then from Equation (12)

$$\|T(f)\|_\infty < \infty, \text{which implies that } T(f) \in B(X).$$

On other hand if $g \in B(X)$, then

$$\|T(f) - T(g)\|_\infty = \max\left\{\left\|T_{ij(i)k(i)}^{\left(\alpha_{ij(i)k(i)},\beta_{ij(i)k(i)}\right)}(f) - T_{ij(i)k(i)}^{\left(\alpha_{ij(i)k(i)},\beta_{ij(i)k(i)}\right)}(g)\right\|_\infty, i \in I\right\}$$

$$\leq \max\left\{\left|\alpha_{ij(i)k(i)}\right|\|f - g\|_\infty, i \in I\right\}, \text{ From(8)}$$

$$= \max\left\{\left|\alpha_{ij(i)k(i)}\right|, i \in I\right\}\|f - g\|_\infty,$$

Hence

$$\|T(f) - T(g)\|_\infty \leq \alpha\|f - g\|_\infty, \quad \forall f, g \in B(X)$$

where $\alpha = \max_{i \in I}|\alpha_{ij(i)k(i)}|$. ∎

The following corollary is a consequent from the previous results, and is an important tool in our work.

Corollary 2.10 If $\alpha < 1$ then, there is a unique image $f_T \in (B(X), \|.\|_\infty)$, such that

$$f_T = T(f_T) = \lim_{n\to\infty} T^{on}(f), \forall f \in B(X).$$

Proof:

From Theorem 2.9, T is lipstchiz on $B(X)$ with α factor, immediately if $\alpha < 1$, T will be contractive on $B(X)$, then T has a unique fixed point f_T because $B(X)$ is a Banach space.

Let T be an image operator specified by a partition R and the code parameters $\left(j(i), k(i), \alpha_{ij(i)k(i)}, \beta_{ij(i)k(i)}\right), i \in I$. Then, T is a fractal operator if $\alpha_{ij(i)k(i),} < 1, \forall i \in I$, where $R_i$is encoded by $\left(j(i), k(i), \alpha_{ij(i)k(i)}, \beta_{ij(i)k(i)}\right)$.

3. Fractal Image Compression Using a Filtered Transform

We propose that the range pool can be divided into two pools, one contains the ranges thatare constants (or nearly constant), then encode its ranges by their suitable parameters, however run the fractal algorithm to encode the others.

Let T be an image operator that specified by a partition $R = \{R_i\}_{i\in I}$ and the code parameters $\left(j(i), k(i), \alpha_{ij(i)k(i)}, \beta_{ij(i)k(i)}\right)$, $i \in I$. For simplicity we refer to $D_{ij(i)}, \alpha_{ij(i)k(i)}, \beta_{ij(i)k(i)}$ as D_i, α_i, β_i respectively, and T_i is the local mapping which defined previously as $T_{ij(i)k(i)}$, where T_i is associated with $D_i, \alpha_i, \beta_i, \forall i \in I$.

We say that f is aconstant image if $\exists c \in \mathbb{R}$ s.t $f(x) = c$; $\forall x \in X$. Let $\hat{I}$ be the set of the indices of the constant sub images of f, i.e.

$$\hat{I} = \{i \in I: (\exists c_i)(c_i \in \mathbb{R})(f_i(x) = c_i)(\forall x \in X)\}.$$

Now let's modify the fractal components to be

$$\dot{T}_i(f)(x) = \begin{cases} c_i & , i \in \hat{I} \\ (T_i f)(x) & \text{otherwise} \end{cases}$$

$\forall x \in X$.

Then the modified fractal operator can be defined as:

$$\hat{T}(f)(x) = \hat{T}_i(f)(x), x \in R_i. \quad (14)$$

The modified operator $\hat{T}$ inherits many of its mathematical properties from the originalone T. For example, it is easy to verify that $\|\hat{T}(f)\|_\infty = \max\left\{\|\hat{T}_i(f)\|_\infty, i \in I\right\}$. Hence, $\hat{T}$ is well defined on $B(X)$, which is a complete metric space, and this makes sense of the following theorems.

Theorem 3.1 $\hat{T}$ is lipschitz, with lipschitz constant

$$\hat{\alpha} = \max\{\|\alpha_i\|, i \in I \backslash \hat{I}\}.$$

Proof:

Let $f, g \in \boldsymbol{B}(\boldsymbol{X})$**,** then

$$\|\hat{T}(f) - \hat{T}(g)\|_\infty = \max\left\{\|\hat{T}_i(f) - \hat{T}_i(g)\|_\infty, i \in I\right\}$$

$$= \max\left\{\max\left\{\|\hat{T}_i(f) - \hat{T}_i(g)\|_\infty, i \in I \backslash \hat{I}\right\}, \max\{\|c_i - c_i\|_\infty, i \in I\}\right\}$$

$$\leq \max\left\{\alpha_i \|f - g\|_\infty, i \in I \backslash \hat{I}\right\}$$

$$\leq \alpha \|f - g\|_\infty. \blacksquare$$

Furthermore, ifα and $\hat{\alpha}$ are the lipschitz factors of T and $\hat{T}$ respectively, then

$$\hat{\alpha} \leq \alpha, \quad (15)$$

since:

$$\hat{\alpha} = \max_{i \in I \backslash \hat{I}} |\alpha_i| \leq \max\left(\max_{i \in I \backslash \hat{I}} |\alpha_i|, \max_{i \in \hat{I}} |\alpha_i|\right) = \max_{i \in I} |\alpha_i| = \alpha.$$

From (15), if $\alpha < 1$ then $\hat{\alpha} < 1$, but it is possible for $\hat{\alpha}$ to be less than one although α is not, i.e., the modified operator $\hat{T}$ may be contractive whenever the base one T is not.

3.1 The Proposed Algorithm

The proposed algorithm is a modification for the standard one and this modification is done depending on the mathematical setting that stated and proved in the previous section.

We will modify the standard encoding algorithm to be:

1. get an image $f^*, \epsilon > 0$
2. construct $R = \{R_i\}_{i \in I}$such that:

   $\cup_{R_i \in R} R_i = X$, $R_i \cap R_j = \emptyset, i \neq j$,

   $C = \{c_i = avr(f_i)\}_{i \in I}$, $V = \{v_i = varr(f_i)\}_{i \in I}$, and $D = \left\{D_j\right\}_{j \in j}$
3. Extract some $R_i \in R$,do:
4. if the variance $v_i < \epsilon$, then
   - (a) store the average ci to be the code parameter of R_i,
   - (b) put the range block index i in $\hat{I}$,
   - (c) omit $R_i$from R, and go to step 6.
5. find the fractal code parameters by the standard algorithm and store them as thecode of R_i.
6. if $R \neq \emptyset$,then go to step 3,
7. end.

Next, the fractal decoding algorithm, which is significantly simpler and faster than the encoding process, is described. The decoding algorithm will be:

1. Let $f_0$be any initial image, $m = 0$,

2. for each range block, $R_i;\ i \in I$,

(a) if $i \in \hat{I}$, then set the grey-level values in $R_i$to c_i, else,

(b) if $i \notin \hat{I}$, set the grey-level values in R_i,by the standard fractal decoding algorithm,

3. Put $f_{m+1} = \hat{T} f_m, if\ \|f_{l+1} - f_l\| \cong 0$then stop else repeat these steps with $m = m + 1$.

3.2 <u>Experimental Results and Discussion</u>

The implementation of the introduced algorithms is done to show that the experimentalresults support the theoretical results proved previously. The computer simulations havebeen carried out in Visual C# environment on Pentium Dual CPU with 1.73 GHz and2.00GB RAM.

The following results are given for a 256×256 pixel grey-scale images. The resultsare different for different images depending on the characteristics of each one as shown inTable1.

Table 1 compares fractal image compression results where the standard scheme is thatintroduced in section 2.6 and the filtered is given above. Error measures are computedby comparing the original bitmap image with an image that has been decoded using 20iterations (more iterations would have resulted in slightly smaller errors) with tolerance0.5, PSNR is the peak signal-to-noise ratio.

M	**Encoding Time (sec)**		**PSNR (dB)**		**CR (pbb)**	
Image	Standard FIC	Filtered FIC	FIC	Filtered FIC	FIC	Filtered FIC
Constant	127	27	44.4	54.4	4.2:1	14.2:1
Lenna	198	187	30.1	28.8	4.2:1	4.1:1
Airplane	191	166	30.4	29.6	4.2:1	3.5:1

Table 1: Results of the Standard algorithm vs the Filtered algorithm

In our experiments, gray level images of size 256×256 have been considered, each oneis partitioned into range blocks of size 4×4 and 8×8 domain blocks. According ouralgorithms, the range blocks pool $\boldsymbol{R}$, the set of all 4×4 pixel non-overlapping sub-squaresof the image, will contain 4096 squares. And the domain blocks pool $\boldsymbol{D}$, the set of all 8×8pixel sub-squares of the image(may be overlapping), will contain $249^2 = 62,001$ squares.

Let $\boldsymbol{R_1}, \boldsymbol{R_2}, \ldots, \boldsymbol{R_{4096}}$ to be the elements of $\boldsymbol{R}$, and $\boldsymbol{D_1}, \boldsymbol{D_2}, \ldots, \boldsymbol{D_{62001}}$to be the elements of D.

In the standard approach, for each $\boldsymbol{R_i}$, scan $\boldsymbol{D}$ to find a $\boldsymbol{D_j} \in \boldsymbol{D}$ whose correspondingimage is the most similar image to that of $\boldsymbol{R_i}$. Considering the 8 isometries, there are$8(62{,}001) = 496008$ squares that should be compared with each of the 4096 range squares.

Moreover, each square in $\boldsymbol{D}$ is 4 times in pixels as an $\boldsymbol{R_i}$, so the nonoverlapping $\mathbf{2} \times \mathbf{2}$ sub-squares of $\boldsymbol{D_j}$ are averaged corresponding to each pixel of $\boldsymbol{R_i}$.

However, in the modified approach, the search step is omitted for some special rangeblocks which reduces the encoding time for the images that contain a significant number ofthese special types.

In the fractal code, each range block is encoded by a transformation and the parametersof the transformation should be stored to recover the corresponding range block. Eachtransformation requires 31 bits to store its parameters, 16 bits for the position of $D_i(8\ bits\ in\ the\ x\ direction\ and\ 8\ bits\ in\ the\ y\ direction)$, 7 bits for β_i, 5 bits for $\alpha_i$and 3 bitsfor the isometry mapping τ_i. There is no need to store the position of $R_i$because it canbe determined implicitly from the transformations ordering. Thus, the fractal code of thewhole image requires approximately 15872 bytes of the storage, whereas the original imagerequires 65,536 bytes of storage, giving a compression ratio of 4.2:1.

In our system each range block needs one bit to be marked as a smooth region ornot, and 8 bits to code the smooth region to store its average, unless its code will be thecorresponding fractal local transformation parameters. Hence, the total space cost of therange block $R_i$is 9 bits if $i \in \hat{I}$, otherwise it costs 32 bits.

REFERENCES

[1] Barnsley, M.F. & A.D. Sloan ,Chaostic compression, Computer Graphics World Nov (1987), 107-108.

[2] M. Barnsley and A. Sloan, A better way to compress images. Byte, vol. 13, no. 1,(1988), 215223.

[3] Barnsley, M.F.& L.P. Hurd, Fractal Image Compression, AK, Peters, Boston, (1993).). Peters, Wellesley.

[4] P. Sheridan, "Spiral Architecture for Machine Vision", PhD Thesis, Faculty of IT, University of Technology, Sydney, 1996.

[5] M. Kramm, Compression of image clusters using Karhunen Loeve transformations, in Electronic Imaging, Human Vision, vol. XII, no. 6492, (2007), 101106.

[6] Rajeshri, R , Yashwant S.C. , "Escape Time Fractals of Inverse Tangent Function", International Journal of Computer & Organization Trends,Vol.3, Issue 3, (2013) 16-21.

[7] Bouboulis, P., P.L. Dalla & V. Drakopoulos, 'Image compression using recurrent bivariate fractal interpolation surfaces.' Internat. J. Bifur. Chaos 16(7):(2006), 2063-2071.

[8] Satyendra, K.P., Munshi, Y., and Arunima, Fracint formula for overlaying fractals, Journal of Information Systems and Communication 3, Issue 1, (2012), 347-352.

[9] T. A. Ramstad and S. Lepsy, "Block-based attractor coding: Potential and comparison to vector quantization," in Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, (1993), 1504-1508.

[10] Fisher, Y. , Fractal Image Compression-Theory and Application. New York: Springer-Verlag (1994).

[11] D. Saupe, R. Hamzaoui, and H. Hartenstein, "Fractal image compression an introductory overview," in Fractal Models for Image Synthesis, Compression and 114 Analysis, ACM SIGGRAPH'96 Course Notes 27 (D. Saupe and J. Hart, eds.), (New Orleans, Louisiana), (1996).

[12] Jacquin, A.E. , 'Image coding based on a fractal theory of iterated contractive image transformations.' IEEE Trans. Image processing 1:(1992), 18-30.

[13] G. Davis, "Self-quantized wavelet subtrees: A wavelet-based theory for fractal image compression," in Proceedings DCC'95 Data Compression Conference (J. A. Storer and M. Cohn, eds.), IEEE Computer Society Press, March (1995).

[14] D. Saupe, "A new view of fractal image compression as convolution transform coding," IEEE Signal Processing Letters, vol. 3, no. 7, (1996).

[15] Rudin, W. "Real and Complex analysis", McGraw-Hill, New York, (1972).

[16] Mohsen, Ghazel, "Adaptive Fractal and Wavelet Image Denoising", Phd Thesis, Waterloo, (2004).

[17] Aida A.Ovanesslan, "Singular Iterated Function System Theory And Its Impact on Fractalization of Gray-scaled Images", Phd Thesis, University of Baghdad, (1996).

[18] Kenneth Falconer, "FRACTAL GEOMETRY Mathematical Foundations and Applications", Wiley and jhons, Enland, (2003).

[19] Barnsley, M. F. and Ervin, V. and Hardin, D. and Lancaster, J., "Solution of an inverse problem for fractals and other sets", Proc. Natl. Acad. Sci. USA 83, 1975-1977, (1986).

[20] John E. Hutchinson, "Introduction To Mathematical Analysis", School of Mathematical Sciences, (1995).

[21] Rashad Al-Jaw_, "Solving the Inverse Problem of Fractals Using Neural Networks",Phd Thesis, University of Baghdad, (2003).

[22] Barnsley, M., "Super fractals", cambridge university press, cambridge, (2006).

[23] Barnsley, M., "Fractals Everywhere", 2ed, Academic Press, San Diego, (1993).

[24] Barnsley, M. F. and Demko,S. "Iterated function systems and the global construction of fractals", Roy.Soc, 243-275, London, Septemper,(1985).

[25] Hutchinson, J.E., "fractals and self-similarity", Mathematics, Indiana university, Vol. 30, no. 4, 713-747, (1981).

[26] SimonK. Alexander, "Two- and Three-Dimensional Coding Schemes for Wavelet and Fractal-Wavelet Image Compression", Msc Thesis, Waterloo, Ontario, Canada, (2001).

[27] Kominek, J., "Advances in fractal compression for multimedia applications", Multimedia Systems, Vol 5, no 4, 255-270, Springer, (1997).

[28] Huawu, Lin, FRACTAL IMAGE COMPRESSION USING PYRAMIDS, Phd Thesis, University of Toronto , Canada , (1997).

Authors' biography

Rashad A. Al-Jawfi is an associate professor in the Department of Mathematics and computer science at Ibb University, Yemen.

He received his PhD in Fractal Geometry and Neural Networks.

He has more than 12 papers published in international journals. He is the author of two books under publishing. He is serving as a member of the editorial board of three international Journals, and reviewer at five international journals.

Baligh M. Al-Helali

Adil M. Ahmed