



Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/14152
DOI URL: <http://dx.doi.org/10.21474/IJAR01/14152>



RESEARCH ARTICLE

APPLICATION OF FIXED POINT THEOREM FOR DIGITAL IMAGES

Rita Pal

Thakur Shyamnarayan Degree College, Mumbai University, Maharashtra.

Manuscript Info

Manuscript History

Received: 30 November 2021

Final Accepted: 31 December 2021

Published: January 2022

Key words:-

Banach Contraction Principle, Digital Metric Space, Digital Image, Increasing Sequence, Complete Digital Metric Space, Finite Sequence, Continuity, Contractive Type Mapping

Abstract

In this research paper, we prove some fixed point theorems for digital images. The paper's main goal is to present another generalisation of the well-known Banach contraction principle for digital images. The fundamental concepts of digital images are discussed, as is an application of the fixed point theorem to digital image compression and fractal image compression. The digital contractive type mappings in digital metric space are introduced, and the uniqueness of fixed points in digital metric space is proved.

Copy Right, IJAR, 2022., All rights reserved.

Introduction:-

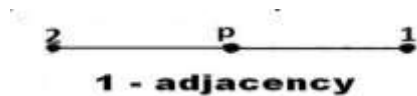
The basic concept of metric fixed point theory is the Banach contraction principle, which states that "If S is a mapping from a complete metric space (X, d) into itself satisfying $d(Sx, Sy) \leq \alpha d(x, y)$ for all $x, y \in X$, where $0 \leq \alpha < 1$, Then S has a unique fixed point." [12]. The Banach contraction principle also provides the existence and uniqueness of fixed points, as well as methods for obtaining approximate fixed points. It was generalised several times by using various types of minimal commutative along with continuity as one of the mappings. We include the following steps for common fixed point: (a) Commutative type condition, (b) Completeness of one or more mappings' range space, (c) Relationship between mapping ranges, (iv) Continuity of one or more mappings, (v) Contractive type condition. The study of spaces with the fixed point property is central to topological fixed point theory. The study of spaces with the fixed point property is central to topological fixed point theory. Furthermore, topology is the study of geometric problems that do not rely solely on the exact shape of the objects, but rather on their interaction with a space. In topology, we generally consider an infinite number of points in a point's arbitrary small neighbourhood. Rosenfeld introduced the concept of digital topology to consider a finite number of points in a neighbourhood. [13]. In fact, digital topology is the study of geometric and topological properties of digital images through the use of geometric and algebraic topology. Digital topology is the study of the topological properties of image arrays. The findings lay the groundwork for image processing operations such as image thinning, border following, contour filling, and object counting. Digital images have been used in a variety of applications, including image processing and computer graphics. Furthermore, digital topology serves as the mathematical foundation for image processing operations. In addition, digital topology is a developing field in both 2D and 3D digital images. Ege and Karaca defined a digital metric space and applied the well-known Banach Contraction Principle to digital images. In this paper, we introduce contractions and contractive mappings in digital metric spaces. We prove the existence and uniqueness of fixed points in digital metric space. Hutchinson [17] initiated the theory of iterated function systems (IFS). Barnsley first recognized the potential of the fractals for the image compression and applied the theory of IFS. Barnsley published his book *Fractals Everywhere* [18]. He also published a paper on fractal image compression [19]. This research activity attracted many researchers in applied mathematics and computers towards fractals.

Preliminaries

Let X be a subset of Z^n for a positive integer n where Z^n is the set of lattice points in the n -dimensional Euclidean Space and β represent an adjacency relation for the members of X . A digital image consists of (X, β) .

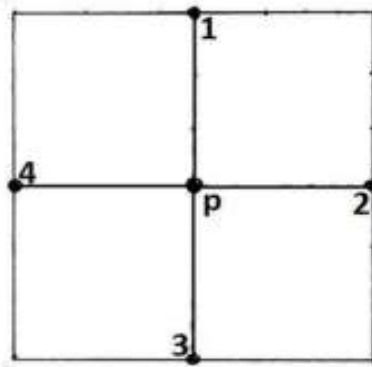
Definition 2.1: Let β, n be positive integers, $1 \leq \beta \leq n$ and p, q be two distinct points $p = (p_1, p_2, \dots, p_n)$, $q = (q_1, q_2, \dots, q_n) \in Z^n$. p and q are β adjacent if there are at most β indices i such that $|p_i - q_i| < 1$ and for all other indices j such that $|p_j - q_j| \neq 1$, $p_j = q_j$. The following statements can be obtained from definition 2.1 For a given $p \in Z^n$, the number of points $q \in Z^n$ which are β adjacent to p is denoted by $k = k(\beta, n)$. It may be noted that $k(\beta, n)$ is independent of p .

2.1.1 If $p \in Z$ (i.e. $n = 1$) then β can take only one value $\beta = 1$. In this case, $k(1, 1) = 2$, since $p-1$ & $p+1$ are the only points 1-adjacent to p in Z . Thus, $k = k(1, 1) = 2$ and q is 1-adjacent to p if and only if $|p-q| = 1$.

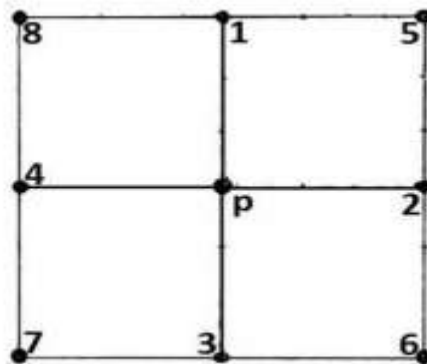


2.1.2 If $p \in Z^2$ (i.e. $n = 2$) then β can take values $\beta = 1, 2$. When $\beta = 1$, the points 1-adjacent to $p = (p_1, p_2)$ are $(p \pm 1, p_2)$, $(p_1, p_2 \pm 1)$. Thus, the number of points 1-adjacent to $p = (p_1, p_2)$ is 4, so that $k = k(1, 2) = 4$. (fig. (a))

When $\beta = 2$, the points 2-adjacent to $p = (p_1, p_2)$ are $(p_1 \pm 1, p_2)$, $(p_1, p_2 \pm 1)$, $(p_1 \pm 1, p_2 \pm 1)$. Thus, the number of points 2-adjacent to $p = (p_1, p_2)$ is 8, so that $k = k(2, 2) = 8$. (fig. (b))



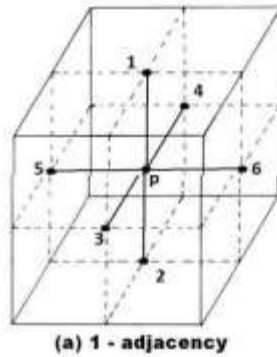
(a) 1 - adjacency



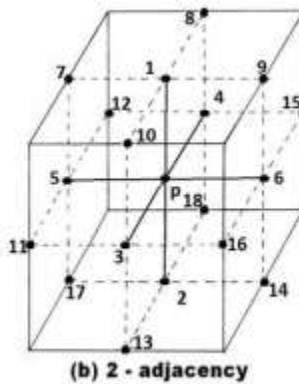
(b) 2 - adjacency

2.1.3 If $p \in Z^3$ (i.e. $n = 3$) then β can take values $\beta = 1, 2, 3$. When $\beta = 1$, the points 1-adjacent to $p = (p_1, p_2, p_3)$ are $(p_1 \pm 1, p_2, p_3)$, $(p_1, p_2 \pm 1, p_3)$, $(p_1, p_2, p_3 \pm 1)$.

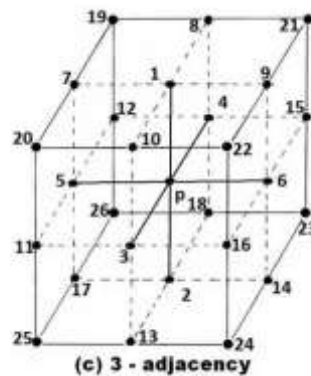
Thus, the number of points 1-adjacent to p is 6, so that $k = k(1, 3) = 6$. (Fig. (a))



When $\beta = 2$, the points 2-adjacent to $p = (p_1, p_2, p_3)$ are $(p_1 \pm 1, p_2, p_3)$, $(p_1, p_2 \pm 1, p_3)$, $(p_1, p_2, p_3 \pm 1)$, $(p_1 \pm 1, p_2, p_3)$, $(p_1 \pm 1, p_2, p_3 \pm 1)$, $(p_1, p_2 \pm 1, p_3 \pm 1)$. Thus, number of points 2-adjacent to p is 18, so that $k = k(2, 3) = 18$. (fig. (b))



When $\beta = 3$, the points 3-adjacent to $p = (p_1, p_2, p_3)$ are $(p_1 \pm 1, p_2, p_3)$, $(p_1, p_2 \pm 1, p_3)$, $(p_1, p_2, p_3 \pm 1)$, $(p_1 \pm 1, p_2 \pm 1, p_3)$, $(p_1 \pm 1, p_2, p_3 \pm 1)$, $(p_1, p_2 \pm 1, p_3 \pm 1)$, $(p_1 \pm 1, p_2 \pm 1, p_3 \pm 1)$. Thus, the number of points 3-adjacent to p is 26, so that $k = k(3, 3) = 26$. (fig. (c))



In general to study n -D digital image, if $1 \leq \beta \leq n$ then $k = k(\beta, n)$ is given by the following formula [12]

$$k(\beta, n) = \sum_{i=n-\beta}^{n-1} 2^{n-i} c_i^n$$

Where $c_i^n = n! / (n-i)! i!$

Suppose X is a non-empty subset of Z^n , $1 \leq \beta \leq n$, $k = k(\beta, n)$

Then (X, β) is called a digital image with β -adjacency¹³. We also say that (X, β) is called n -D digital image . [8]

Definition 2.2: Let $X \subset \mathbb{Z}^n$, d is the Euclidean metric on \mathbb{Z}^n . (X, d) is a metric space. Suppose (X, β) is a digital image with β -adjacency, then (X, d, β) is called a digital metric space. [14]

Definition 2.3: A sequence $\{x_n\}$ of points of the digital metric space (X, d, β) is a Cauchy sequence if there is $M \in \mathbb{N}$ such that, $d(x_n, x_m) < 1$ for all $n, m > M$.

Theorem 2.1: For a digital metric space (X, d, β) , if a sequence $\{x_m\} \subset X \subset \mathbb{Z}^n$ is a Cauchy sequence, there is M, N such that for all $n, m > M$, we have $x_n = x_m$.

Definition 2.4: A sequence $\{x_n\}$ of points of a digital metric space (X, d, β) converges to a limit $L \in X$, if for all $\epsilon > 0$, there is $M \in \mathbb{N}$ such that $d(x_n, L) < \epsilon$ for all $n > M$.

Proposition 2.1 : A sequence $\{x_n\}$ of points of a digital metric space (X, d, β) converges to a limit $L \in X$ if there is $M \in \mathbb{N}$ such that $x_n = L$, for all $n > M$. (i.e. $x_n = x_{n+1} = x_{n+2} = \dots = L$).

Definition 2.5: A digital metric space (X, d, β) is complete if any Cauchy sequence $\{x_n\}$ converges to a point L of (X, d, β) .

Theorem 2.2 : A digital metric space (X, d, β) is complete.

Definition 2.6 : Let (X, d, β) be a digital metric space and $T : (X, d, \beta) \rightarrow (X, d, \beta)$ be a self-map. If there exists $\lambda \in [0, 1)$ such that, $d(Tx, Ty) \leq \lambda d(x, y)$, for all $x, y \in X$, then T is called a contraction map.

Proposition 2.2: Every digital contraction map $T : (X, d, \beta) \rightarrow (X, d, \beta)$ is β -continuous (Digital continuous).
Lemma 2.1: Let $X \subset \mathbb{Z}^n$ and (X, d, β) be digital metric space. Then there does not exist a sequence $\{x_m\}$ of distinct elements in X , such that $d(x_{m+1}, x_m) < d(x_m, x_{m+1})$, for $m = 1, 2, 3, \dots$ i.e. there exist a finite sequence $\{x_m\}$. [15]

Main Result

First we introduce a notion.

Notion 3.1: Let $\Theta = \{\theta : [0, \infty) \rightarrow [0, \infty)\}$ be such that θ is increasing, $\theta(t) < \sqrt{t}$ for $t > 0$, $\theta(t) = 0$ iff $t = 0$.

Definition 3.1: Suppose (X, d, β) is a digital metric space, $T : X \rightarrow X$ and $\theta \in \Theta$. Suppose $d(Tx, Ty) \leq \theta(d(x, y))$, $x, y \in X$. Then T is called a digital θ -contraction.

Now we prove a fixed point theorem on θ -contraction.

Theorem 3.1 Suppose (X, d, β) is a digital metric space, $T : X \rightarrow X$ and $\theta \in \Theta$. Suppose $d(Tx, Ty) \leq \theta(d(x, y))$, for all $x, y \in X$. T is called a digital θ -contraction. Then T has unique fixed point.

Proof: Let $x_0 \in X$ and suppose $x_{n+1} = Tx_n$, for $n = 0, 1, 2, 3, \dots$. We may suppose that $x_n \neq x_{n+1}$, for $n = 0, 1, 2, \dots$.

Otherwise x_n is a fixed point.

$$\begin{aligned} \text{Now, } d(x_{n+1}, x_n) &= d(Tx_n, Tx_{n-1}) \\ &\leq \theta(d(x_n, x_{n-1})) \end{aligned}$$

$$\begin{aligned} &= \theta(d(x_{n-1}, x_n)) \\ &< \sqrt{d(x_{n-1}, x_n)} \\ &\leq d(x_{n-1}, x_n) \end{aligned}$$

Therefore $d(x_{n+1}, x_n) < d(x_{n-1}, x_n)$, ($\because x_n \neq x_{n-1}$)

Therefore $d(x_{n+1}, x_n)$ is strictly decreasing sequence. Therefore $x_n = x_{n+1}$, for large n . By Lemma (2.1)

Therefore x_n is a fixed point of T , for large n . Uniqueness of fixed point of T .

Suppose u and v are fixed points of T .

$$\begin{aligned} \text{Then, } d(u, v) &= d(Tu, Tv) \\ &\leq \theta(d(u, v)) \end{aligned}$$

$$< \sqrt{d(u, v)}$$

$$\leq d(u, v)$$

It is a contradiction, if $u \neq v$.

Therefore $u = v$.

Hence T has unique fixed point.

Corollary 3.1 (Banach Contraction Principle in Digital Metric Spaces)

Let (X, d, β) be a digital metric space and $T : X \rightarrow X$ be such that $d(Tx, Ty) \leq \lambda d(x, y)$ for all $x, y \in X$ and for some $\lambda \in [0, 1)$. Then, T has unique fixed point.

Proof: Take $\theta(t) = \lambda t$ in Theorem 3.1. Then we get the result

Theorem 3.2 Let (X, d, β) be a digital metric space and $T : X \rightarrow X$ such that $d(Tx, Ty) < \mu(x, y)$, for all $x, y \in X, x \neq y$. Where $\mu(x, y) = \max\{\frac{1}{2} [d(y, Ty) \frac{1+d(x, Tx)}{1+d(x, y)} + d(Tx, Ty) + d(x, y)], d(x, Tx) \frac{1+d(y, Ty)}{1+d(Tx, Ty)}\}$. Then T has unique fixed point.

Proof: Let $x_{n+1} = Tx_n$ for $n = 0, 1, 2, \dots$

Suppose $x_n \neq x_{n+1}$ for $n = 0, 1, 2, \dots$

Otherwise x_n is a fixed point.

Now,

$$d(x_n, x_{n+1})$$

$$= d(x_{n-1}, x_n) < \mu(x_{n-1}, x_n)$$

$$= \max\{\frac{1}{2}[d(x_n, Tx_n) \frac{1+d(x_{n-1}, Tx_{n-1})}{1+d(x_{n-1}, x_n)} + d(Tx_{n-1}, Tx_n) + d(x_{n-1}, x_n)], d(x_{n-1}, Tx_{n-1}) \frac{1+d(x_n, Tx_n)}{1+d(Tx_{n-1}, Tx_n)}\}$$

$$= \max\{\frac{1}{2}[d(x_n, x_{n+1}) \frac{1+d(x_{n-1}, x_n)}{1+d(x_{n-1}, x_n)} + d(x_n, x_{n+1}) + d(x_{n-1}, x_n)], d(x_{n-1}, x_n) \frac{1+d(x_n, x_{n+1})}{1+d(x_n, x_{n+1})}\}$$

$$\leq \max\{\frac{1}{2}[d(x_n, x_{n+1}) + d(x_{n-1}, x_{n+1})], d(x_{n-1}, x_n)\}$$

$$\leq \max\{\frac{1}{2}[d(x_{n-1}, x_n), d(x_{n-1}, x_n)]\}$$

$$= d(x_{n-1}, x_n)$$

Hence, $d(x_n, x_{n+1}) < d(x_{n-1}, x_n)$ for $n = 0, 1, 2, \dots$

Hence by Lemma (2.1)

T has a fixed point.

Uniqueness of fixed point of T

Let u and v are two fixed points of T .

Therefore $Tu = u$ and $Tv = v$

Now,

$$d(u, v) = d(Tu, Tv)$$

$$< \mu(u, v)$$

$$= \max\{\frac{1}{2}[d(v, Tv) \frac{1+d(u, Tu)}{1+d(u, v)} + d(Tu, Tv) + d(u, v)], d(u, Tu) \frac{1+d(v, Tv)}{1+d(Tu, Tv)}\}$$

$$= \max\{\frac{1}{2}[d(v, v) \frac{1+d(u, v)}{1+d(u, v)} + d(u, v) + d(u, v)], d(u, u) \frac{1+d(v, v)}{1+d(u, v)}\}$$

$$= \max\{\frac{1}{2}[d(u, v) + d(u, v), 0]\}$$

$$= \max\{d(u, v), 0\}$$

$$= d(u, v)$$

$$= d(u, v) < d(u, v), \text{ for } u \neq v \text{ which is contradiction.}$$

Therefore $u = v$

Hence there exists unique fixed point.

This completes the proof.

An Application of Fixed Point Theorems to Digital Images

Many mathematicians are interested in fractal geometry. Mandelbrot was the first to use the term "fractal." A fractal is a geometric shape in which each part is a reduced copy of the whole. The following are some fractal examples.



Fig 1:- Sierpinski Triangle.

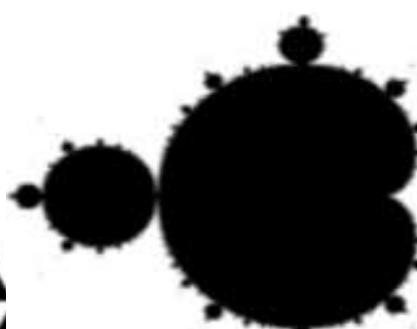


Fig 2:-Mandelbrot Set.

Fractals are used to approximate many real-world objects, such as coastlines, mountains, trees, and clouds. Mandelbrot's book, *The Fractal Geometry of Nature* [16], sparked widespread interest.

Fractal image compression techniques are varied, but they represent only a small portion of all compression methods available. If the basic concept of an image compression technique is to utilise the self-similarities that naturally occur in many photos, it is considered fractal. A part of an image can frequently be found that, if altered in some way, would fit into another part of the same image.

Fractals are often described as self-similar objects; that is where the "fractal" part comes from. Benoit B. Mandelbrot developed the term "fractal" from the Latin word *fract* or *fractus*, which means "broken" or "uneven." Instead than attempting to define a fractal precisely, one may take an alternative approach to the problem. We might build a list of features that characterise fractals, as Falconer states in "Fractal geometry mathematical foundation and application"[20]. A fractal set F may not have all of the following features, but at least some of them:

1. F is self-similar on some scales;
2. F is detailed on all scales; and
3. F 's fractal dimension (specified in some way) does not have to be an integer in general.
4. F is usually described in terms of a simple algorithm.

A typical example of a fractal is the Sierpinski triangle. It's made by starting with an equilateral triangle (T_0) and locating the triangle's midpoints on each side. A new triangle is made by drawing lines between these locations, and this new triangle is removed by removing the lines. Inside the original triangle, we now have three additional equilateral triangles (see Figure 3). T_1 stands for triangles. The next iteration is formed by repeating this process for each of the new triangles, and so on. The Sierpinski triangle is constructed by repeating it an unlimited number of times. The set is similar to the Cantor set.

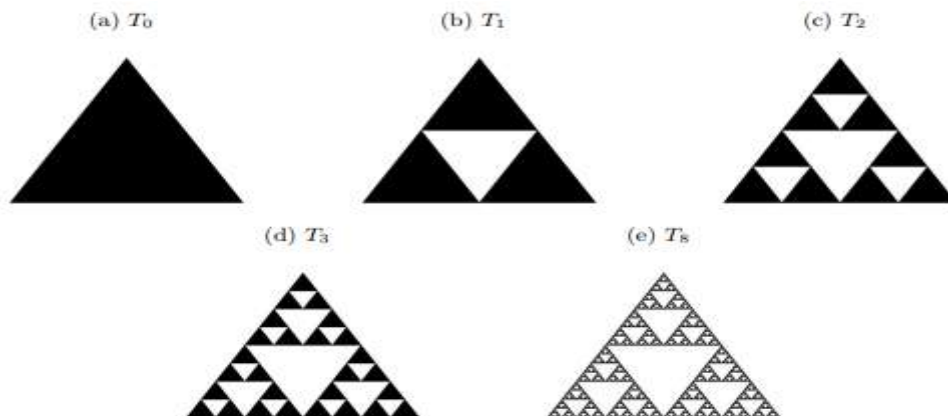


Figure 3:- Construction of the Sierpinski Triangle.

The area and circumference of the Sierpinski triangle is quite interesting as well. We start of by investigating the area of T_n .

If T_0 has side length 1 it has height $\sqrt{1^2 - \frac{1^2}{4}} = \frac{\sqrt{3}}{2}$.

$n=0$: The area a_0 of the first triangle T_0 then is $\frac{1}{2} \cdot \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{4}$.

$n=1$: Since we removed a fourth of the triangle T_0 to create T_1 the area of T_1 is

$$a_1 = a_0 - \frac{1}{4} a_0 = \frac{3\sqrt{3}}{16}.$$

$n=2$: T_2 was created by removing a fourth of each triangle in T_1 , therefore the area of T_2 is

$$a_2 = \frac{3^2}{4^2} a_0 = \frac{9\sqrt{3}}{64}.$$

In general the area of T_n is $a_n = \frac{3^n}{4^n} a_0 = \frac{3^n \sqrt{3}}{4^{n+1}}$ for $n \geq 0$.

Hence, the area of the Sierpinski triangle is $\lim_{n \rightarrow \infty} a_n = 0$

The circumference of T_n , where we also count the circumference of each hole, can be found by:

1. $n=0$ Each side of T_0 is of length 1, which yields the circumference is $l_0 = 3$.
2. $n=1$ T_1 consists of three equilateral triangles each with $\frac{1}{2}$ the side length of T_0 . This gives us the circumference $l_1 = 3 \cdot 3 \cdot \frac{1}{2} = \frac{9}{2}$.
3. $n=2$ Since it is a repeating pattern T_2 consists of three times as many triangles as T_1 and each have $\frac{1}{2}$ the side length of a triangle in T_1 . Hence the circumference of $T_2 = 3 \cdot \frac{9}{2} = \frac{27}{2}$.

Here T consists of 3^n triangles with side length $\frac{1}{2^n}$. The circumference of T_n then is $l_n = \frac{3 \cdot 3^n}{2^n} = \frac{3^{n+1}}{2^n}$, which tends to $+\infty$ as $n \rightarrow +\infty$. Therefore, the Sierpinski triangle has area = 0 but infinite circumference.

This seemingly strange phenomenon is explained by the third property of fractals mentioned above. A common illustrative notion of fractal dimension is the so-called box dimension of a set. It is a scaling relationship in which the number of boxes required to cover a set scales with the side length of the boxes. Figure 4 shows that the Sierpinski triangle is covered by 4 boxes with side length $\frac{1}{2}$ in (a) and by 12 boxes with side length $\frac{1}{4}$ in (b). In general if N boxes with side length ϵ is needed to cover the Sierpinski triangle then it takes $3N$ boxes with side length $\frac{\epsilon}{2}$ to cover it. Since the number of boxes increase by 2^d , where $d = \frac{\log 3}{\log 2}$, we say that this number d is the box dimension of the Sierpinski triangle.

In many cases, the box dimension is equal to many other notions of dimension, so d is referred to as the set's fractal dimension. With this in mind, it appears more likely that the Sierpinski triangle has no area but is more than just a curve, as its dimension is a number between 1 and 2. There are other definitions of fractal dimensions besides the box dimension, such as the Hausdorff dimension, which is more mathematically convenient. The fractal dimension is an intriguing topic, but it is not the subject of this thesis. More information, however, can be found in [20] for those who are interested.

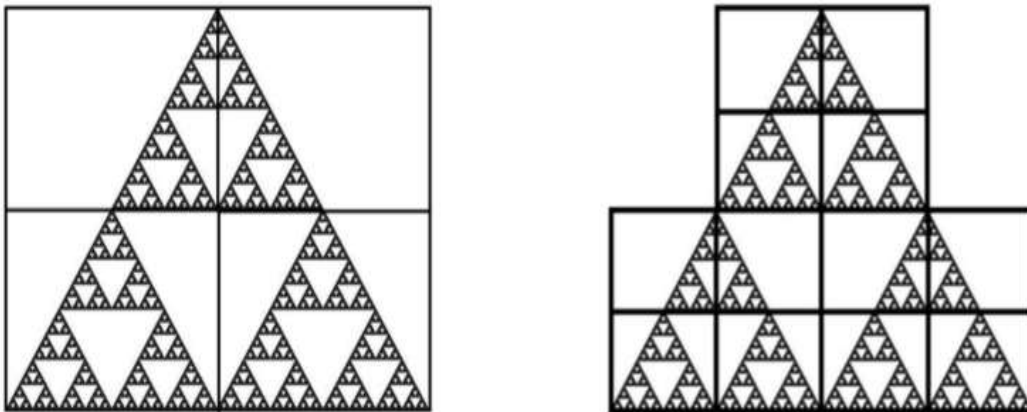


Figure 4:- Illustration of boxes it take to cover the Sierpinski triangle.

To begin, we allow X to represent any set. By defining X abstractly, we can discuss a wide range of different sets. We'll work with sets of sets and, to a smaller extent, sets of images later on. However, for readers who are unfamiliar with the concepts described below, it may be helpful to think of X as \mathbb{R} or \mathbb{R}^2 .

Definition 4.1. A metric space (X, d) is a set X together with a real-valued function $d: X \times X \rightarrow \mathbb{R}$ such that for any $x, y, z \in X$, the following holds:

1. $d(x, y) = 0 \Leftrightarrow x = y$;
2. $0 < d(x, y) < \infty$, if $x \neq y$;
3. $d(x, y) = d(y, x)$;
4. $d(x, y) \leq d(x, z) + d(z, y)$.

Thus a function d is called a metric.

It's important to note that a space doesn't have to have its own metric. For example, we may measure the distance between two points $x = (x_1, x_2)$ and $y = (y_1, y_2)$ in the space \mathbb{R}^2 by,

$$d_1(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

This real-valued function clearly meets the requirements of Definition 4.1 (i)-(iv), indicating that it is a metric. On the other hand, we have the "Taxicab metric," which is occasionally referred to as such.

$$d_2(x, y) = |x_1 - y_1| + |x_2 - y_2|$$

It satisfies the properties as well. As a result, both d_1 and d_2 are metrics in the \mathbb{R}^2 space. d_1 in its most general form is

$$d_e(x, y) = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \dots + (x_n - y_n)^2)}$$

and This metric d_e will be referred to as the Euclidean metric.

Definition 4.2. A sequence $\{x_n\}$ in a metric space (X, d) is said to converge to a point $x \in X$, if given any $\epsilon > 0$, there exists an $N \in \mathbb{N}$ such that $d(x_n, x) < \epsilon$ whenever $n > N$.

Definition 4.3. A sequence $\{x_n\}_{n=1}^{\infty}$ in a metric space (X, d) is called a Cauchy sequence if, for any $\epsilon > 0$ there is an $N \in \mathbb{N}$ such that $d(x_n, x_m) < \epsilon \forall n, m > N$.

In other words, when one proceeds further down a Cauchy sequence, the points become closer and closer together. They do not, however, need to arrive at a specific location in space X . Consider the metric space (\mathbb{Q}, d) , where \mathbb{Q} denotes rational numbers and d denotes the Euclidean metric. The integer e is not in \mathbb{Q} , so the sequence $a = (1 + \frac{1}{n})^n$ converges to e . As a result, the following definition is appropriate:

Definition 4.4. A metric space (X, d) is complete if every Cauchy sequence in X converges in X .

The concept of compact subsets is another crucial topic in the development of fractal theory. To grasp the concept of compact subsets, one must first recall what a subsequence is. Consider a sequence $\{x_n\}$; a subsequence $\{x_{n_k}\}$ can be created from $\{x_n\}$ by deleting some or all of the components while keeping the order of the remaining elements the same. For example, the sequence $1/2, 1/4, 1/6, \dots$ is the even denominator subsequence of the sequence $\{1/n\}_{n=1}^{\infty}$, where $n = 1, 1/2, 1/3, \dots$

Definition 4.5. Let $E \subset X$ be a subset of the metric space X in (X, d) . If every infinite sequence in E possesses a subsequence that converges to an element in E , then E is said to be compact.

The concept of compact subsets might be a bit odd to the reader who has not encountered it before. Therefore we will state some other definitions regarding subsets of metric spaces, together with a theorem, that will help with the intuition of compact subsets.

Definition 4.6. A subset E of a metric space (X, d) is open if, for each point $p \in E$ there is some $r > 0$ such that $\{q \in X : d(p, q) < r\}$ is contained in E .

Definition 3.6. A subset E of a metric space (X, d) is closed if the complement of E , denoted E^c , is open.

Remark. A closed set contains all its limit points.

Definition 4.7. A subset E of a metric space (X, d) is bounded if there exists a real number $M > 0$ and a point $q \in X$ such that $d(p, q) < M$ for all $p \in E$.

Theorem 4.1. Any compact subset $K \subset X$ in a metric space (X, d) is closed and bounded.

Converse of Theorem 4.1 is not true in general, but we do obtain equivalence when $X = \mathbb{R}^k$ and the Euclidean metric are used. The Heine–Borel Theorem is the form of the theorem that holds when $X = \mathbb{R}^k$. The fractals outlined above, as well as those built later, can be regarded as closed and bounded subsets of \mathbb{R}^2 . (or \mathbb{R} in the case of the Cantor set).

Definition 4.8. An Iterated Function System (IFS) is a finite set of contraction mappings on a complete metric space (X, d) ,

$$\{w_i : X \rightarrow X \mid i = 1, 2, \dots, N\}.$$

Each contraction mapping w_i has a corresponding contractivity factor c_i . An alternative notation for the same IFS is,

$$\{X; w_1, w_2, \dots, w_N\}.$$

The two mappings w_1 and w_2 , which in fact are contraction mappings, could be used to construct the Cantor set.

We will now state a theorem that generalises this concept for any given IFS.

Recall that if E is a subset of X and $f : X \rightarrow X$ is a function, then we define $f(E) = \{f(x) : x \in E\}$.

Theorem 4.2. Let $\{X; w_i, i = 1, 2, \dots, N\}$ be an IFS with contractivity factor $c = \max\{c_1, c_2, \dots, c_N\}$.

Define the transformation $W : H(X) \rightarrow H(X)$ by

$$W(B) = \bigcup_{i=1}^N w_i(B) \text{ for all } B \in H(X). \text{ Then:}$$

- (i) $W(B)$ is a contraction mapping with contractivity factor c with respect to the Hausdorff metric;
- (ii) Its unique fixed point $A \in H(X)$ with $A = \bigcup_{i=1}^N W(A) = w_i(A)$ is given by
 $A = \lim_{n \rightarrow \infty} W^n(B)$ for any $B \in H(X)$. The fixed point $A \in H(X)$ is called the attractor of the IFS.

Example 4.1 (Sierpinski triangle). Earlier, we built the Sierpinski triangle by removing the middle part of a triangle and repeating the process. However, the Sierpinski triangle can also be represented using an iterative function system. Start off with a solid triangle T_0 . Then T_1 is constructed with the use of three functions, these three functions are affine transformations. Each transformation scales the triangle by a half and places each scaled down triangle (in form of translations) in the corners of T_0 .

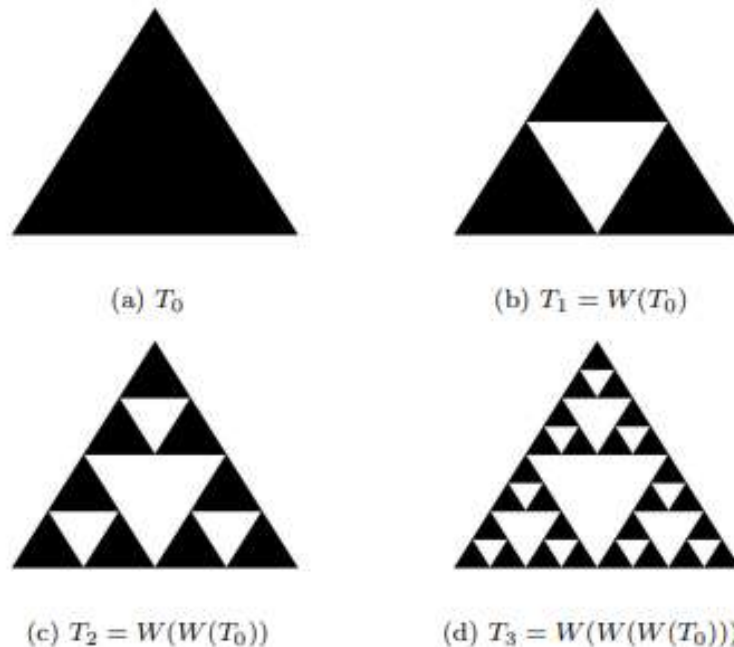


Figure 5:- Using the IFS to construct the Sierpinski triangle.

The corresponding IFS is given by $\{R^2 : w_1, w_2, w_3\}$ where the contractive transformations w_1, w_2 and w_3 are given by $w_1(x_1, x_2) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$$w_2(x_1, x_2) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}$$

$$w_3(x_1, x_2) = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1/4 \\ \frac{\sqrt{3}}{2} \end{pmatrix}$$

The attractor T of this IFS is the Sierpinski triangle and is given by $T = \lim_{n \rightarrow \infty} W^n(T_0)$.

Theorem 4.2 states that the attractor of an IFS is unique and given an IFS it does not matter what the initial set is, in the end iterates of W will tend to its attractor. To answer the inverse, how one should go about finding an IFS for a given attractor the Collage Theorem proves useful.

Theorem 4.3 (The Collage Theorem, (Barnsley)). Let (X, d) be a complete metric space. Let $L \in H(X)$ and $\epsilon \geq 0$ be given. Choose an IFS $\{X : w_1, w_2, \dots, w_n\}$ with contractivity factor $0 \leq c < 1$, such that

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

where h is the Hausdorff metric. Then,

$$h(L, A) \leq \frac{h(L, \cup_{i=1}^n w_i(L))}{1-c} \leq \frac{\epsilon}{1-c} \text{ for all } L \in H(X)$$

where A is the attractor of the IFS.

This results follows from the contraction mapping theorem. If one considers f to be a contraction mapping with unique fixed point x_f , and let $x \in X$ be such that $d(x, f(x)) < \epsilon$ for a given $\epsilon > 0$. Then

$$\begin{aligned} d(x, x_f) &= d(x, f(x_f)) \leq d(x, f(x)) + d(f(x), f(x_f)) \\ &\leq d(x, f(x)) + c \cdot d(x, x_f). \end{aligned}$$

Hence,

$$d(x, x_f) \leq \frac{d(x, f(x))}{1-c} \leq \frac{\epsilon}{1-c}$$

In Example 4.1 above we identify the IFS by analysing which transformations are needed for constructing the Sierpinski triangle.

Fractal Image Compression

In the physical world, a photo is simply a piece of paper with an Inc. on it, but in the digital world, a photo is a collection of small logical units called pixels. The number of pixels in a digital image is referred to as its resolution. The resolution of an image is $n \times m$ if it is n pixels wide and m pixels high. A digital image with a resolution of $n \times m$ can be thought of as a $n \times m$ matrix, with each entry representing a pixel. The colour of a pixel is determined by the value of an entry, the pixel value. The number of distinct colours that a pixel can represent is determined by the number of bits per pixel used. For example, 8-bit colour allows for the display of 2^8 different colours. A bit (short for "binary digit") has a single binary value of "0" or "1," and can only answer "yes" or "no" questions. Each pixel in a grayscale digital image is frequently 8 bits wide, which means that a grayscale digital image with a resolution of 1024×1024 requires $1024 \cdot 1024 \cdot 8 = 8.4 \cdot 10^6$ bits to store. The storage size required to save an image will be referred to as the image's memory size. Memory is typically measured in bytes, with one byte equalling eight bits.

Despite the fact that high-speed Internet access is spreading around the world and connection speeds are increasing, it is still limited. The time it takes to send a data file is determined not only by the connection speed, but also by the file's memory size. As a result, sending a high-resolution image or a collection of images may still take some time. Compressing the images reduces the amount of data that must be transferred, as well as the time it will take. But how can images be compressed? The human eye's sensitivity to a variety of information losses is an important characteristic. In other words, an image can be altered in ways that the human eye is incapable of detecting. If there is a lot of redundant data that doesn't affect the "big picture," the data can be greatly compressed. Lossy compression methods are those that lose some information during the compression process, whereas lossless compression methods do not lose any original data.

The basic idea behind fractal image compression is to store (also known as encode) images as a set of transformations. To be useful, there must be a method to decompress the image, i.e., a method to reconstruct the image from the stored information. The decompression (or decoding) process entails repeatedly applying the transform to an arbitrary starting image, resulting in an image that is either the original or, in most cases, very similar to it. Instead of pixel values, each image in Figure 7 can be saved as a collection of affine transformations. If the numbers in the transformations are of the commonly used data type "float," then each number has a memory size of 32 bits. Storing the tree as a collection of transformations, for example, only requires 4 transformations x 6 numbers x 32 bits per number = 768 bits. Storing it as a collection of pixels, on the other hand, necessitates $512 \times 512 \times 1 = 262,144$ bits for the resolution of 512×512 . (Since it is only black and white, we only need 1 bit to store the color). With this in mind, one might wonder if it is possible to find a small number of affine transformations that represent any given image. The answer is simply no, because a natural image is not exactly self-similar, but it is also not completely devoid of self-similarity. As previously stated, when looking at an image, one may notice a portion of it that, when scaled and rotated, fits into another portion of the same image. Self-similarities of this type can be found in most images of faces, cars, mountains, and so on. To make use of these similarities, we must partition the image in some way and compare the bits and pieces.

Metric Spaces of Images

To use the main results from the previous sections, we need a complete metric space. A grayscale image mathematical model is a function $f: S \rightarrow G$, where the domain S represents points on paper and the range G is the colour of the points. For the sake of simplicity, we will assume that S is a closed rectangular region in \mathbb{R}^2 and that $f(x, y) \in G$ exists for every $(x, y) \in S$, where G represents a closed interval of grayscale values ranging from black to white. Using the function $f(x, y)$, we can generate a 3D-graph where the height represents the grey level at each point (x, y) on the paper. We define the metric in order to be able to say anything about the differences or distance between two images, f and g .

$$d^*(f, g) = \sqrt{\int |f(x, y) - g(x, y)|^2 dx dy}$$

If we define F to be the space of real-valued square-integrable functions $f: S \rightarrow G$, then F together with the metric d^* forms a complete metric space. [6]

Recall that $W: F \rightarrow F$ is a contraction mapping if for some constant $c, 0 \leq c < 1$

$$d^*(W(f), W(g)) \leq c \cdot d^*(f, g),$$

where c is called the contractivity factor of W . Then, by the Contraction mapping theorem, there exists a unique fixed point $f_w \in F$ satisfying $W(f_w) = f_w$.

We can state the Collage Theorem for grayscale images in the following way. Let f be a grayscale image and assume that $W: F \rightarrow F$ is a contraction mapping such that

$$d^*(f, W(f)) \leq \epsilon.$$

Then

$$d^*(f, f_w) \leq \frac{\epsilon}{1-c}$$

where c is the contractivity factor of W , f_w is its fixed point and $W^n(f_0) \rightarrow f_w \approx f$ for any initial image f_0 .

The Fractal Block Coding Algorithm

The Collage Theorem is the underlying principle of fractal image compression. The theorem guarantees that we can find a fractal representation of an image if we can find a contraction mapping on F . In light of this, A. Jacquin presented the basic fractal block coding algorithm in 1992 [4]. The algorithm begins by segmenting the image into

m non-overlapping range blocks R_i ($1 \leq i \leq m$). These range blocks can be thought of as functions $R_i : R_i \rightarrow G$ from the "spatial part" $R_i \subset S$ of the range block R_i to G . Then another partition of the image is made, this time into n non-overlapping domain blocks D_j ($1 \leq j \leq n$). In the same way $D_j : D_j \rightarrow G$ are functions from the "spatial part" D_j of D_j to G . Each domain block has in general twice the side length of the range blocks. An illustrative example can be seen in Figure 8.

Given these two partitions the fractal block coding algorithm search, for each range block R_i , the best match amongst the domain blocks. Since it is unlikely all range blocks have a good match amongst the domain blocks we are allowed to modify said domain blocks. This can be done by shifting the grayscale value of the entire domain block by a constant β , and scaling each grayscale value by a constant α . In the end we have, for each range block R_i a matching domain block $D_{j(i)}$ together with the α_i and β_i values for the match. The list of triples ("index of domain block", α, β) will form the encoding of the image.

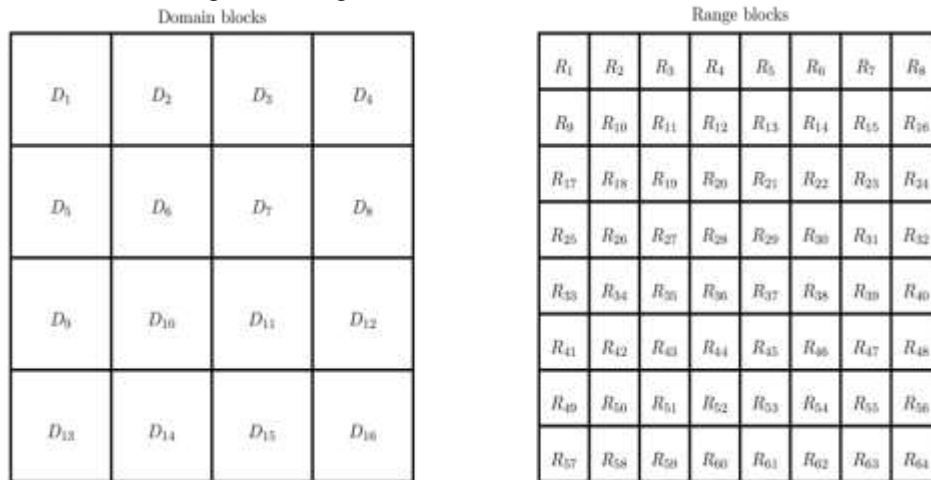


Figure 6:- Example of 64 range blocks of size $B \times B$ and 16 Domain Blocks of size $2B \times 2B$.

Let α_i and β_i denote the best grayscale scaling and shift respectively and let $v_i : D_{j(i)} \rightarrow R_i$ denote the unique affine map of the form $v_i(x, y) = \frac{1}{2}(x, y) + (a, b)$, mapping $D_{j(i)}$ onto R_i for $1 \leq i \leq m$. Then we have the following theorem:

Theorem 5.1. Let $W_i : F \rightarrow F$, for $1 \leq i \leq m$, be defined as

$$W_i(f)(x, y) = \begin{cases} \alpha_i f(v_i^{-1}(x, y)) + \beta_i & \text{if } (x, y) \in R_i \\ 0 & \text{if } (x, y) \in S \setminus R_i \end{cases}$$

Then W_i is a contraction mapping with respect to d^* , if $|\alpha_i| < 2$.

Proof :

$$\begin{aligned} (d^*(W_i(f), W_i(g)))^2 &= \int_{R_i} |W_i(f)(x, y) - W_i(g)(x, y)|^2 dx dy \\ &= \int \alpha_i^2 |f(v_i^{-1}(x, y)) - g(v_i^{-1}(x, y))|^2 dx dy \\ &= \frac{\alpha_i^2}{4} \int_{D_i} |f(x, y) - g(x, y)|^2 dx dy \\ &\leq \frac{\alpha_i^2}{4} \int_{D_i} |f(x, y) - g(x, y)|^2 dx dy \\ &= \frac{\alpha_i^2}{4} (d^*(f, g))^2 \end{aligned}$$

This means that if $\frac{\alpha_i^2}{4} < 1$, then W_i is a contraction.

Since the R_i 's ($1 \leq i \leq m$) forms a partition of S , we can define $W : F \rightarrow F$ by

$$W(f)(x, y) = \sum_{i=1}^m W_i(f)(x, y) = \alpha_i f(v_i^{-1}(x, y)) + \beta_i \quad \text{if } (x, y) \in R_i$$

If we choose α_i such that W_i is a contraction for all $i \in \{1, 2, \dots, m\}$, then by the Contraction Mapping theorem, through iteratively applying W_i to any starting image f we will recover the fixed point f_{W_i} . If we now define f_W as the sum of all f_{W_i} , we have the following theorem.

Theorem 5.2. Suppose $c := \max |\alpha_i/2| < 1$. Let f_{W_i} denote the unique fixed point of the contraction mapping W_i ($i = 1, \dots, m$), and let $f_W = \sum_{i=1}^m f_{W_i}$. If $W(f) = \sum_{i=1}^m w_i(f)$ for any $f \in F$, then there exists a constant γ depending on f such that

$$d^*(W^{\circ j}(f), f_W) \leq \gamma \cdot c^j$$

Proof :

$$\begin{aligned} d^*(W^{\circ j}(f), f_W) &\leq \sum_{i=1}^m d^*(W_i^{\circ j}(f), f_{W_i}) \\ &\leq \sum_{i=1}^m \left(\frac{\max_i |\alpha_i|}{2} \right)^j d^*(f, f_{W_i}) \\ &= \left(\max_i \left| \frac{\alpha_i}{2} \right| \right)^j \sum_{i=1}^m d^*(f, f_{W_i}) \end{aligned}$$

Fractal Compression of Grayscale Digital Images

A grayscale digital image can be represented by a function

$\tilde{f}: \{1, 2, \dots, n\} \times \{1, 2, \dots, m\} \rightarrow \{0, 1, \dots, 255\}$, and as mentioned earlier, when working with digital images (which are of fixed size, say $n \times m$) we can think of them as matrices. We let $|\tilde{f}_i, j|$ for $i = 1, \dots, n$ and $j = 1, \dots, m$, be a matrix where each entry $\tilde{f}_i, j = \tilde{f}(i, j)$.

This gives us a way to compute the difference of two digital images with the so called rms (root mean square) metric:

$$\text{drms}(\tilde{f}, \tilde{g}) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |\tilde{f}(i, j) - \tilde{g}(i, j)|^2}$$

For the sake of simplicity, we will consider square images or matrices, such that $m = n$ for some $n = 2^p$. Each range- and domain-block now represents a submatrix, but the domain blocks are still twice as large as the range blocks. A domain block and a range block must be the same size in order to be compared.

This is accomplished by averaging the domain block's pixel values and reducing its size to that of a range block. The fractal block coding algorithm for grayscale digital images can now be stated as follows:

Algorithm 1 Fractal Block Coding

- 1: for R_i ($1 \leq i \leq m$) do
- 2: for D_j ($1 \leq j \leq n$) do
- 3: Downscale D_j to match the size of R_i and call it \hat{D}^j .
- 4: Find best α and β for the pair (R_i, \hat{D}^j) using rms.
- 5: Compute the error using rms, and if the error is smaller then for any other domain block, remember the pair along with the α and β .
- 6: end for
- 7: end for

This algorithm is a basic fractal image compression algorithm. This method has many variations and improvements, but the basic idea remains the same. A minor change to improve the match between the range and domain blocks is to rotate and flip the domain block. We chose to call this method the enhanced fractal block coding algorithm.

The image is partitioned in two ways, just like the original algorithm. The first partition is made up of non-overlapping domain blocks, while the second is made up of non-overlapping range blocks (see Figure 6).

Then for each range block R_i we find the domain block together with a transformation that is closest to R_i . The transformation tested for each domain block includes: • Flipping; • Rotating; • Changing contrast and brightness.

Flipping is simply a reflection of the scaled down domain block and the rotations includes rotating the block $0^\circ, 90^\circ, 180^\circ$ or 270° . We have eight variants of each domain block to compare each range block with because we can flip or not flip the domain block and then rotate it in four different ways. Then for each range block R_i we find the

version of the down scaled domain block $\hat{D}_j(i)$, together with a contrast scaling constant α and a brightness controlling constant β , which have the lowest root mean square error. i.e. finding the function and domain block $w_i(D_j(i)) = \alpha \times \text{rotate}(\text{flip}(\hat{D}_j(i))) + \beta$ that minimizes with $\hat{f} = R_i$ and $\hat{g} = w_i(D_j(i))$. The algorithm can be stated as following:

Algorithm 2

Enhanced Fractal Block Coding

```

1: for  $R_i$  ( $1 \leq i \leq m$ ) do
2:   for  $D_j$  ( $1 \leq j \leq n$ ) do
3:     Downscale  $D_j$  to match the size of  $R_i$  and call it  $\hat{D}_j$ .
4:     Generate  $\hat{D}_{j,k}$  ( $k = 1, \dots, 8$ ), the different rotations and flipping of  $\hat{D}_j$ .
5:     Find best  $\alpha$  and  $\beta$  for the pair ( $R_i, \hat{D}_{j,k}$ ) using rms.
6:     Compute the error using rms, and if the error is smaller then for any other  $\hat{D}_{j,k}$ ,
       remember the pair  $R_i, D_j$  along with the rotation, flipping,  $\alpha$  and  $\beta$ .
7:   end for
8: end for

```

The decoding for both algorithms is nearly identical, with the only differences being the saved parameters. We begin by generating an arbitrary image of the same size as the original image, and then apply the transformations corresponding to the saved parameters a fixed number of times iteratively. If we have the restriction $|\alpha| < 2$ we ensure that each transformation is a contraction, and then by Theorem 5.2 the decoded image will be close to the original.

When working with lossy image compression (such as fractal image compression), it can be useful to be able to measure the quality of the decompressed image. One common method is to compute the peak signal-to-noise ratio (PSNR). To do so, we must first calculate the mean square error (MSE). For the original $m \times n$ image f and the lossy compressed image f^* the MSE is defined as:

$$MSE = \frac{1}{m \cdot n} \sum_{i=1}^n \sum_{j=1}^m |f(i, j) - f^*(i, j)|^2$$

Since we are working with 8-bit grayscale images the largest pixel value is 255, given this together with the MSE the PSNR is defined as: $PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right)$

It is important to note that the PSNR only measures the overall difference between two images' pixel values. In other words, it says nothing about how the human eye will perceive image quality.. A high PSNR is considered better as we aim to have a small mean square error between the original and the approximated image.

Implementation and Results the two fractal compression algorithms presented above were implemented in Python. To test and compare the algorithms, experiments on two different types of images were conducted. The image is a QR code with the message "Fractal image compression". The original images can be seen in Figure 7. By Theorem 5.1 we need to restrict $|\alpha| < 2$ to ensure a contraction. However, in practice it is useful to restrict $|\alpha|$ further to reduce the number of iterations needed before reaching the fixed point. This might affect the quality of the image a bit but it guarantees that the sequence of images converge faster and only a few decoding steps are necessary.



Figure 7:- Original image of the QR-code.

QR code with the message “Fractal image compression”.

The memory size of an image compressed by either of the presented algorithms does not depend on the original image per se. What determines the memory size is the number of range- and domain blocks we choose together with the choices of α and β . For the enhanced method we also need to take the rotation and flipping into account. The test images in Figure 9 both have resolution 512×512 , so the memory size after compression will be the same when using the same range block size. Two different sizes of the range blocks were tested and in the first case they were 16×16 with the corresponding domain block size of 32×32 . Since the resolution of the test images are 512×512 we have a total of 256 domain blocks of size 32×32 , so the index of the domain block requires 8 bits. In the second case we have range blocks of size 8×8 and domain blocks of size 16×16 which result in a total of 1024 domain blocks, so here 10 bits are required for the index. The α 's were chosen to be any combination of $c_1 \frac{1}{2} + c_2 \frac{1}{4} + c_3 \frac{1}{8} + c_4 \frac{1}{16}$, where $c_i \in \{0, 1\}$ for $i = 1, 2, 3, 4$. Thus, 4 bits were used representing them. The β 's on the other hand were chosen to take any integer value between -255 and 255 , which is a total of $2^9 - 1$ different values. Therefore, 9 bits were needed in the representation of the β 's.

The image created by the standard fractal block coding algorithm is stored as $1024(8 + 4 + 9) = 21504$ bits = 2688 bytes for the larger block sizes and $4096(10 + 4 + 9) = 94208$ bits = 11776 bytes for the smaller block sizes. As mentioned, the enhanced method also needs to store information about the rotation and flipping of the domain block. 1 bit is used for the flipping and 2 bits are used for the rotation. This results in the total memory size of the images with the larger block sizes as $1024(8 + 1 + 2 + 4 + 9) = 24576$ bits = 3072 bytes and for the smaller block sizes $4096(10 + 1 + 2 + 4 + 9) = 106496$ bits = 13312 bytes. By dividing the memory size of the original images (which is $512 \cdot 512 \cdot 8 = 2097152$ bits = 262144 bytes) with the memory size of the compressed images, we get the compression ratio. For example, the compression ratio of the standard fractal block coding algorithm with range block of size 16×16 is $\frac{262144}{2688} = 97.5$.

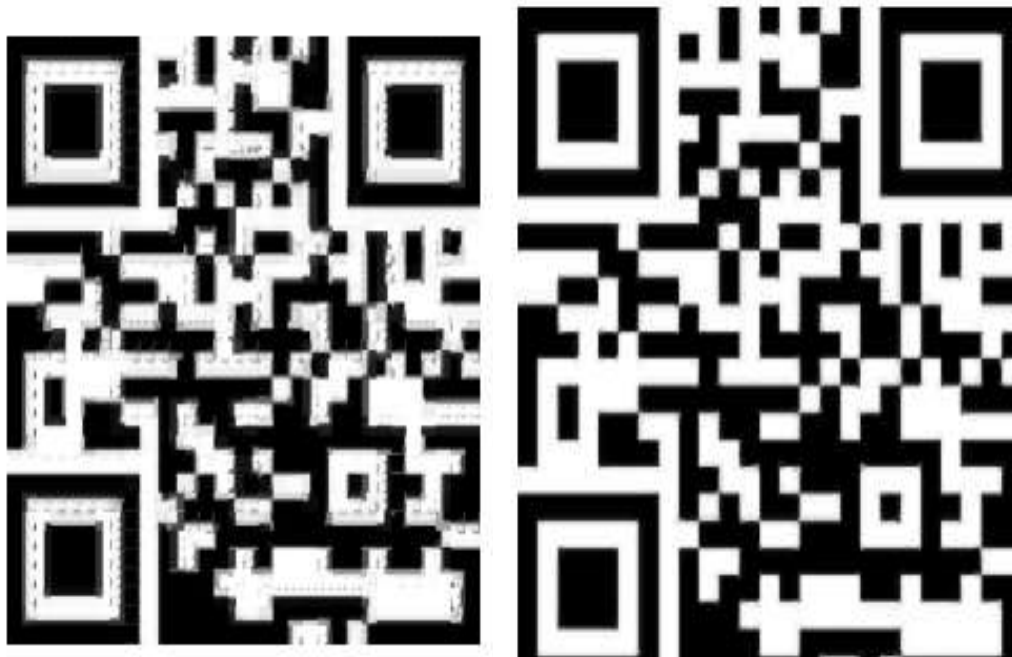
The original image is segmented into parts such that each part is nearly the same as a reduced copy of the original image. The union of all the segments is then close enough to the original image. Thus the images with global self similarity are encoded with extreme efficiency. Unfortunately, a general image is not always globally self similar. In such images, self similarity exists only locally amongst different small parts of it. See the following image.

It has been observed that all the images in nature contain a considerable amount of affine redundancy. The affine redundancy means large segments of the image look like the small segments of the same image. Large segments are known as domain blocks whereas small segments as range blocks. We can find an affine transformation (a combination of rotation, reflection, scaling and shifting transformation) that transforms a domain block to the suitable range block. The parameters of the transformation constitute a fractal code. Thus a range block is approximated by applying an affine transformation on a suitably chosen domain block. Since the mapping reduces the size of the domain block, it is a contractive mapping. Fractal image compression works as follows:

1. The image is partitioned into non-overlapping range blocks. Generally, the partition of an image may have any arbitrary shape (square, rectangles, triangles, quadrilaterals or any polygon).
2. The same image is partitioned into overlapping domain blocks. Domain blocks are larger in size than the range blocks in order to maintain contractive condition.
3. Finally the image is encoded by using a suitable affine transformation which maps a domain block to a best fitted range block.
4. To achieve the decompression, exactly the opposite is done. Inverse affine transform is applied to recover the image. Usually 8 to 9 inverse iterations are applied on the encoded image to decode the image. The iteration starts with any arbitrary image. Successive application of the affine map gives the sequence of images that ultimately converge to a fixed image (by fixed point theorem of Banach).

Conclusion:-

QR code consist of mainly very dark or very bright grayscale colours each "error" is quite notable, especially for the larger block sizes. The 4 fractal images of the QR code in Figure 8 are successfully readable by a QR code scanner. Our purpose is to give the digital version of Banach fixed point theorem by introducing θ -contractive type mapping. These results are the applications of fixed point theory in digital metric space. It will be useful for digital topology and fixed point theory. In the future, we will also use the fixed point theory to solve some problems in digital images.



(a) Standard algorithm with range block size 16×16 .

PSNR: 17.6

Memory size: 2688 bytes.

Compression ratio: 97.5

(b) Standard algorithm with range block size 8×8 .

PSNR: 26.9

Memory size: 11776 bytes.

Compression ratio: 22.3



(c) Enhanced algorithm with range block size 16×16 .

PSNR: 19.2

Memory size: 3072 bytes.

Compression ratio: 85.3

(d) Enhanced algorithm with range block size 8×8 .

PSNR: 28.0

Memory size: 13312 bytes.

Compression ratio: 19.7

Fig 8 :- Results of both fractal compression methods for the QR code with two different range block sizes. The PSNR, memory size and compression ratio is presented for each compressed image.

Finding the best match for each range block is the most computationally intensive part of the algorithms. Because the enhanced method includes 8 variants of each domain block, the encoding part's run-time is approximately 8 times longer than with the current implementation. This is significant because encoding is already a lengthy procedure, especially for smaller block sizes. In theory, the improved method should produce better (or at least comparable) results than the standard fractal block algorithm. The PSNR results back this up, but the memory trade-off may not be worth it in most cases.

Fractal image compression, as previously stated, is a lossy compression method. The Joint Photographic Experts Group, or JPEG, is a more well-known lossy compression method. Even though the fractal block algorithm has a high compression ratio at times, the long encoding time is a significant disadvantage. Other fractal image compression methods attempt to address this shortcoming in order to make fractal compression a more competitive option. However, existing fractal methods are still regarded as time-consuming compression methods when compared to, say, JPEG.

Thus the contractive mappings and fixed point theorem is at the core of the fractal image compression. Important aspect of the fractal image decoding is resolution independence. That means we may compress a 128×128 image, and decompress it to any size, say 64×64 or 256×256 . Fractal image compression produces better reconstructed images than that of JPEG (Joint Photographic Expert Group) technique.

Acknowledgements:-

Authors are thankful to the anonymous referees for their critical remarks. Their valuable suggestions is a key to improved the article, especially the last section.

References:-

- 1 Banach S, "Sur les operations dans les ensembles abstraits et leurs applications equation integrals", Fund. Math., No. 3, pp. 133- 181, 1922.
- 2 Brouwer LEJ, "ubereineindeutige, stetiger Transformation envon Flaachen in Sich", Math. Ann., 69, pp. 176-180, 1910.
- 3 Brouwer LEJ, "uber Abbildungen Von Mannigfaltigkeiten", Math. Ann., 77, pp. 97-115, 1912.
- 4 Kong TY, "A Digital Fundamental Group, Computers and Graphics", 13, pp. 159-166, 1989.
- 5 Boxer L, "Digitally Continuous Functions", Pattern Recognition Letters, 15, pp. 833- 839, 1994.
- 6 Boxer L, "A Classical Constructions for The Digital Fundamental Group", J. Math. Imaging Vis., 10, pp. 51-62, 1999.
- 7 Boxer L, "Properties of Digital Homotopy", J. Math. Imaging Vis., 22, pp. 19-26, 2005.
- 8 Rosenfeld A, "Continuous functions on digital pictures", Pattern Recognition Letters 4, pp. 177-184, 1986.
- 9 Ege O, Karaca I, "Applications of The Lefschetz Number to Digital Images", Bull. Belg. Math. Soc. Simon Stevin, 21, pp. 823- 839, 2010.
- 10 Ege O, Karaca I, "Banach Fixed Point Theorem for Digital Images", J. Nonlinear Sci. Appl. 8, pp. 237-245, 2015.
- 11 Ege O, Karaca I, "Lefschetz Fixed Point Theorem for Digital Images", Fixed Point Theory, Appl., 13 pages, 2013.
- 12 Han S.E, "Non-Product property of the digital fundamental group", Inform. Sci., 171, pp. 73-91, 2005.
- 13 Rosenfeld A, "Digital Topology", Amer. Math. Monthly, 86, pp. 76-87, 1979.
- 14 Han S. E, "Banach fixed point Theorem from the viewpoint of digital topology", J. Nonlinear sci. Appl., 9, pp. 895-905, 2016.
- 15 Sridevi K., M. V. R. Kameshwari And D. M. K. Kiran, "Fixed Point Theorem for Digital Contractive Type Mappings in Digital Metric Space" International Journal of Mathematics Trends and Technology, Vol. 48, No. 3, pp. 159-167, 201
- 16 Mandelbrot B. The fractal geometry of nature. 2nd edition. WH. Freeman and Co. San Francisco; 1982.
- 17 Hutchinson J. Fractals and self similarity. Indiana University Journal of Mathematics. 1981;30:713-747.
- 18 Barnsley MF. Fractals everywhere. 2nd edition. Academic Press. San Diego; 1993.
- 19 Barnsley MF, Sloan AD. A better way to compress images. Byte. 1988;215-223.
- 20 Kenneth Falconer. Fractal Geometry, Mathematical foundations and applications. Wiley , 1990.