Gradient-Based Weighted Central Pixel Matching for Efficient Fractal Image Coding

Nadia M. G. Al-Saidi[1], Suzan J. Obaiys[2(✉)], Arkan J. Mohammed[3], and Yeliz Karaca[4]

[1] Department of Applied Sciences, University of Technology, Baghdad 10066, Iraq
nadia.m.ghanim@uotechnology.edu.iq
[2] Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
suzan@um.edu.my
[3] Department of Mathematics, College of Science, Al-Mustansiriah University, Baghdad, Iraq
drarkanjasim@uomustansiriyah.edu.iq
[4] University of Massachusetts Chan Medical School (UMASS), 55 Lake Avenue North, Worcester, MA 01655, USA
yeliz.karaca@ieee.org

Abstract. In the context of image coding systems, the fractal image encoding and decoding method is a better option due to its capacity for high achievement in compression ratio and resolution independence at any scale. Fractal image coding embodies two major phases. Firstly, the encoding step, where we produce numerical data, called Iterated Function System (IFS) data, from the statistically self-similar input image. Secondly, in the decoding step, we decode IFS data to revert to a fractal image, as an attractor, by using an inverse function. Notwithstanding, the popularity of the fractal encoder in light of the partition iterative function system has deteriorated into nearly out of favour due to its massive encoding time. So, the existing process is still not applicable. This paper proposes a new approach in which block dimensions are adjusted to address the limitation of requiring a high encoding time. The new strategy is designed for efficient pooling selections. It uses the central pixel and its neighbors based on gradient values, giving more weight to pixels with higher gradient magnitudes. This allows the matching process to take into account not just the pixel intensity but the change in intensity across the block, making it more robust to noise or slight variations in pixel values. This study confirms through experiments acceptable improvement in both encoding time and quality of the decoded image. Subsequently, it is considered an enhancement to some of the previous methods in the literature.

Keywords: Fractal image coding · Gradient method · Self-similarity · IFS · Domain and Range partition · Fractal dimension

1 Introduction

People in today's information-rich society are exposed to a vast amount of images on a daily uses, which puts a lot of strain on image transmission and storage. In the early 80s, two popular methods for image coding were Discrete Cosine Transformation (DCT)

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15648, pp. 158–175, 2025.
https://doi.org/10.1007/978-3-031-97000-9_10

[1] and Vector Quantization (VQ) [2]. DCT method uses non-overlapping blocks by partitioning grids where each block is encoded individually, while VQ Methods explore the image pixels directly. Both VQ and DCT methods produced much research in order to achieve targeted image coding, and subsequently, these methods occupied a variety of commercial implementations. Despite this affluence, the method of image coding was extending towards a new direction, particularly fractals and wavelets, which already have achieved appreciable attention [3]. Mathematicians used a mathematical iterative sequence deterministically [4]. This concept led researchers to think of a realistic fractal object and thus, finally, Mandelbrot first rediscovered a realistic visual complex object mathematically and was able to demonstrate it with computational graphics. The Fractal concept deals with non-integer dimensions, while the classical Euclidean dimension exists on integer dimensions [5]. The fractal concept is now used in many fields to analyze objects such as Engineering, Physics, and Biology, security protocols, as well as in different areas of image processing [6–10]. Michael Barnsley [11] was the first who propose that real-life objects or images can be modeled using deterministic fractal objects. This modeling process is performed with the help of a set of iterative affine transformations, he named this set the iterative function systems (IFS) [12] as presented in his famous collage theorem. Nevertheless, the drawback of this theory is that encoding and decoding cannot possible to be automated. Arnaud E. Jacquin first proposed a piece-wise affine contractive transformation in the form of partial transformability of image [3]. On Jacquin's contribution, massive research has taken place till now and comparatively, to some extent, successes have been achieved already and become undeniably popular because of their potential to foster high-resolution re-builder images at very eminent compression ratios [13–18]. Nevertheless, it experienced encoding time complexity due to an immense range of harmonization with a range block to the enormous domain blocks [19]. In order to address this most critical barrier to Fractal Image Coding, several researchers have already introduced methods that reduce the search complexity of the domain pool comparatively [20–23].

Galabov showed how block size affects encoding time and the quality of images, taking 2×2 and 4×4 pixel block sizes [24]. Datta came up with the concept of pixel neighbourhood in the image thinning process and defined the central pixel in 8-neighbours for a binary image [25]. Fisher proposed the division of domain blocks into 72 different classes with combinations of average and variance in four quadrants of the block under consideration [26]. By using mean pixel value and variance of blocks, Xing et al. improved Fisher's layout and achieved 576 classes [27]. Caso et al. used vector variances to transfer the fixed ordering of variances of image blocks and neighbourhood of classes [28]. Wang et al. applied edge information of the blocks for differentiating blocks into a number of classes and gave the result in speed-up of ratio [9]. Fisher and Menlove divided an image into Horizontal-vertical partitioning recursively to form two new rectangles [29]. Hebert and Soundararajan divided a rectangular image into two triangles initially and each of these triangles is subdivided into four triangles by splitting the triangle recursively [2, 30]. To sort image blocks, Han applied a fuzzy structure cataloger [31], and on the other hand, for the same purpose, Al-Saidi et al. used some of the metaheuristic Optimization algorithms to enhance the encoding time [32]. From all the literature above, we notice that all the researchers use approaches that partition the

images into different block sizes, for example: 32×32, 16×16, 8×8, 4×4, or 2×2 and work on entire blocks. As a result, it affects the search process because the pixel intensity values are not symmetric compared to the central pixel of the above-mentioned blocks.

The main contribution in this work is initiated by changing the 2×2 average pooling system into a 3×3 max pooling system of the block-windows; and blocks dimensionality considering 8 neighbours of the central pixel and searching domain blocks depending on the central pixel rather than entire blocks for each of the size of the blocks (3×3). The conventional method uses the intensity value of the central pixel as the index for matching between domain and range blocks. The gradient-based weighting of the central pixel and its surrounding neighbours could be used to expand this. Determine the gradient values (intensity changes) for the eight neighbours surrounding the central pixel rather than just the central pixel. By capturing the direction and magnitude of changes in local pixel values, these gradients can give the value of the central pixel additional context.

The remaining part of the work is arranged as follows: Sect. 2 discusses the mathematical groundwork in detail based on our proposed method. Section 3 shows the development and formation of fractal encoding and decoding through our proposed method. In Sect. 4, we compared our experiment results with other existing work, and finally, we conclude the work in Sect. 5.

2 Mathematical Background

This section presents the mathematical background of feature extraction in digital images. It is a new approach that uses thinning algorithms, which take advantage of the spatial relationships between pixels. The method depends on looking at a block's central pixel as well as its neighbours.

2.1 Pixel Neighborhoods

- Central Pixel (p_c):

Let $p_c = P(i, j)$ be the central pixel in a given block. It is significant because it is encircled by nearby pixels that collectively capture local structural characteristics.

- Orthogonal neighbours ($N_4(p)$):

The neighbors of p_c is defined by the adjacent (vertical and horizontal) pixels such that:

$$N_4(p) = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$$

- Diagonal neighbours ($ND(p)$):

$$ND(p) = \{(i-1, j-1), (i-1, j+1), (i+1, j-1), (i+1, j+1)\}$$

$ND(p)$ is used to define the four diagonal neighbours of p_c

- Complete 8-Neighbor Set ($N_8(p)$):

The full set that used to surround p_c combines the orthogonal and diagonal neighbours, such that:

$$N_8(p) = N_4(p) \cup ND(p),$$

which represents the symmetry and structure around the central pixel.

2.2 The Space Where the Fractal is Generated

To measure variations between image regions, the framework additionally makes use of ideas from metric space theory.

- **Hausdorff Metric:**

When calculating the distance between two sets in a complete and compact metric space, the Hausdorff metric H is utilized. It offers a means of evaluating how similar image subsets under transformation are to one another.

- **Iterative Function Systems (IFS):**

Our approach is based on the following theorems, which summarize the main findings in fractal theory:

Theorem 1: Iterative Function System (IFS). *An Iterative Function System is a finite set of contraction mappings on a complete metric space (X,d), such that,*

$$f_i: X \to X \text{ for } i = 1,2,\ldots,N$$

where each f_i is a contraction, and the IFS system consists of the set $\{f_i\}$. This theorem guarantees that the mappings are applied iteratively, and this theorem ensures convergence to a unique attractor.

Theorem 2. Collage theory uses a set of contractive mappings to approximate an image $\text{Im} \subset X$. If an IFS $F = \{X, f_i\}_{i=1}^{N}$, and a mapping $f_i: X \to X$ exist, and $\epsilon > 0$ with contraction ratios $0 \leq s_i < 1$ then,

$$d_H\left(\text{Im}, \bigcup_{i=1}^{N} f_i(\text{Im})\right) \leq \epsilon$$

Consequently, the bounded distance between the original image and the IFS's attractor A is:

$$d_H(\text{Im}, A) \leq \frac{\epsilon}{1-s}$$

where s is the maximum contraction factor. This outcome guarantees that we can use the attractor produced by the IFS to closely resemble the original image by reducing the collage error.

2.3 Gradient Principles

A. Gradient Computation: Calculate the gradient in both vertical and horizontal directions for every 3 × 3 block. To find the gradient magnitudes and directions for each neighbour of the central pixel, we can use the Sobel or Prewitt filter operators.

B. Weighted Matching: Based on the magnitude of these gradients, give the central pixel and its neighbours weights after you have the gradient information. The matching score will be higher for blocks with similar gradient patterns, which indicate similar edge or texture features.

C. Block Matching: Compare the weighted sum of gradients between domain and range blocks rather than the individual pixel values. This method can lessen the sensitivity to minute changes in pixel values and enables more reliable matching, particularly in pictures with subtle textures or gradients.

By adding the directional information of the surrounding pixels, this technique enhances central pixel indexing and may result in faster and more accurate block matching, particularly in images with high-frequency details.

2.4 Application of IFS and Collage Theory.

We can take a set of $M(i, j, [g_p])$ in a metric space, (X, d), where $X \in \mathbb{R}^n$, and $\mathbb{R}^n$ denotes n-dimensional real space. (i, j) is a pixel location in spatial coordinates, and g_p is the pixel value. The set $M(i, j, [g_p])$ is closed and bounded, so according to [11], M is a compact set and $(M(i, j, [g_p]), d)$ compact metric space and all sequences will converge to its limit point in $(M(i, j, [g_p]), d)$ [11]. But when we measure the distance between two sets, the symmetric characteristics of a metric space are not met. We must therefore evaluate distances using different metrics, as the Hausdorff Metric. Fractal objects can exist in a Hausdorff space [11]. So, a set M in a Hausdorff metric space is a complete metric space. Any set M will converge to its limit point using Hutchinson's operator, resulting in a unique attractor [12]. Barnsley used the Hutchinson operator naming *IFS,* which is a set of successive contraction mappings on a complete metric space with contractive factors $0 < s < 1$, and mathematically, $A = W_n(A)$ is an attractor (Theorems 1 and 2). The set A is the same as $n = 1$, the union of the image for all points in A under W_n [6]. This demonstrates how a certain IFS maps onto a sequence of A_{i-1} in a complete metric space, for $i = 1, 2, \ldots$, which converge to its distinctive limit point. Now, for the same IFS, we take a set $\mathrm{L} \subset (\mathrm{X}, d_H)$ in a complete metric space where, $d_H(L, W_n(L)) < \varepsilon$ is the Hausdorff distance. This distance is measured between L and A, for $n = 1$, it is less than (1-s) [5]. This approach did not work on statistically self-similar natural images in Machine Vision. In 1992, however, Jacquin understood that the natural image is not exclusively self-similar. That is, any subset of the image is not similar to the whole image; rather, any subset of the image is similar in many scales to another subset of the image [19]. This idea gives him a new dimension to contribute to the method of Barnsley [3], and thus he proposes a partitioned iterated function system (PIFS), which is a block-wise contraction mapping of a square size of blocks in place of the whole image in Theorem 2. The steps of this method are presented as follows:

1. Jacquin took range and domain blocks from the same set with the size R_j and D_i.
2. $D_i = 4\ R_j$ where $D(2r, 2r)$ is the array of overlapping blocks and $B\ (r, r)$ is the array of non-overlapping blocks.
3. The size of D_i is contracted to the size of B_j homogeneously, immediately before the eight isometric mappings are applied on D_i, to be $r \times r$.
4. Every block is contracted, that is, every 2×2 set of pixels is reduced to a (1×1) set of $(2r \times 2r)$, (4 pixels). For this purpose, Jacquin's method used 2×2 average pooling.

So far, all researchers have used these basics. Image is encoded to calculate PIFS, and it is decoded to get back the image (the fractal attractor) using the collage theorem [5].

3 The Proposed Method

We have already observed a variety in block size while researchers work in terms of a pixel array. In this method, (3×3) square window of blocks is chosen due to its consistency and symmetric with the central pixel. However, when the 8-isometric transformation on the window of size (3×3) pixels is applied, the central pixel remains intact. This central pixel helps to match between domain blocks and range blocks.

3.1 Partitioning of Domain and Range Pools

The uniform partitions of an image of size M in a complete metric space (X, d) are constructed such that the union of all subsets in M must be equal to the entire set M. In the first step of the process, we have taken two versions of block pools for the same image $(M \times M)$ as a domain pool and range pool. Then the size of each domain block $D_i = 3r \times 3r$ pixels is captured as a sub-block, while $R_i = r \times r$ for the range.

3.2 The Geometrical Transformations

In the geometrical transformations, we apply Homothety and Similitude iteratively on a set M in the complete metric space, and thus we can achieve a unique fixed point or attractor, for more details see [18] and [5]. Pixel values must be transformed to perform operations like geometric transformations and intensity adjustments in digital image processing. The original technique involved mapping the intensity values of pixels within a grayscale image's spatial coordinate system. Assume that the image is represented by a 2D matrix $f(i, j) = g_p$, where g_p refers to the pixel intensity value at (i, j), and $g_p \in [0, 2^b - 1]$, b is the number of bits in each pixel (in the standard grayscale images, it equals 8).

A color image is defined by three separate channels: Red, Green, and Blue (RGB), each channel has its own intensity value for every pixel. A pixel at (i, j) is defined by the RGB values as $f_R(i, j), f_G(i, j), f_B(i, j)$, ranging from 0 to 255 (for 8-bit images). To apply this transformation to color images, we should handle the three distinct channels.

The mapping function must be modified while preserving the overall transformation for every pixel. The transformation for pixel intensities can be given as follows:

$$I_R(z) = s_R z_R + g_R$$
$$I_G(z) = s_G z_G + g_G$$
$$I_B(z) = s_B z_B + g_B$$

where z_R, z_G, z_B are the pixel intensity values in the Red, Green, and Blue channels, and s_R, s_G, s_B and g_R, g_G, g_B represent the scaling and adjustment factors for each channel.

The coordinates of the pixels in a color image do not change when undergoing geometric transformations like translation, rotation, or scaling. These transformation rules can be used to change the intensity values of each channel. To maintain the integrity of the image's color balance, it is important to make sure that the changes made to the pixel values are the same in each of the three channels.

3.2.1 The Contraction Mapping

In image processing, we operate on the general pooling to mitigate variance, trim the complexity of computation and obtain low-level features from the neighbourhood. We use max pooling because it can reduce 50% of the time over average pooling. In this stage, we reduced the $\kappa r \times \kappa r$ sub-block size to $(r \times r)$ size of the sub-block. A homogeneous contraction mapping is used, which means every $\kappa \times \kappa = 3 \times 3$ pixels block is condensed to 1×1, as shown in Fig. 1. This can be done with either average pooling or max pooling. We proposed max pooling, and subsequently, we reduced the pooling time by 50%, maintaining the average PSNR is almost 27 dB as shown in Tables 2 and 4.

3.2.2 Isometric Transformation

In geometric isometric transformation, the affine transformation is a linear mapping method that preserves points, straight lines, and planes. They consist of four reflections (R_f) and four rotations (R_o) that can be applied on domain blocks to its mapped range blocks under some conditions for balancing the brightness and contrast of the image. The affine transformation w is defined as:

$$w\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{pmatrix}\begin{pmatrix} i \\ j \\ k \end{pmatrix} + \begin{pmatrix} e \\ f \\ g \end{pmatrix} \tag{1}$$

$$R_f = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} \tag{2}$$

$$R_o = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3}$$

Figure 1 shows how the 8-isometrics are equally divided in 4-reflections using Eq. (2), where $\cos 2\theta = a$, $b = \sin 2\theta = c$ and $-\cos(2\theta) = d$. By turns the values of $\theta = -45°$, $0°$, $45°$, $90°$, and in 4-rotations using the Eq. (3) where $a = \cos(\theta) = d$, $b = -\sin(\theta) = c$, where $\theta = 0°$, $90°$, $180°$, $-90°$.

3.3 Pixel Values Transformations

In the geometrical aspect of gray-scale digital images, there are spatial coordinates (i, j) such that $f\ (i, j) = g_p$, where $g_p \in [0, 2^b - 1 \mid b = 1, 2,\ldots,8]$. Here $[i \times j]$ is a *2D* image plane in a complete metric space. For every (i, j), we have a value of pixel intensity g_p under the function $w(i, j) = g_p$. Even after all 8-isometric transformations are applied on (3×3) windows, the central pixel neither changes its position nor its values according to the function $f(i, j) = g_p$. Windows except κ x κ, where $\kappa = n$ and $n \in$ N is any odd value, do not have these characteristics. Nevertheless, we use $\kappa = 3$ to comply with 8-neighbours, which is a new idea in this field. So, in this spatial coordinate window of domain blocks, the process of mapping into the range blocks is possible because of translation vectors. Hence, to maintain the total mapping, we need to introduce the third dimension, which will attain the value of g_p and consequently, the function will be rearranged from Eq. 1 as:

$$\Rightarrow f\begin{pmatrix} i \\ j \\ z \end{pmatrix} = \begin{pmatrix} ai + bj + e \\ ci + dj + f \\ sz + g \end{pmatrix}$$

It is clear that, we have an equation for depicting the intensity function $I\ (z) = sz + g$ with variable z and $I\ (z)$ are the domain and range block intensity, respectively. We can estimate a linear regression line $\hat{I}(z) = +\hat{s}qz$ and thus we get the error,

$$E = \sum \{I(z) - \hat{I}(z)\}^2 = \sum \left\{I(z) - g_q - \hat{s}_q z\right\}^2$$

where $z = f(i, j)$ which returns g_q and to minimize error, we can take partial derivatives with respect to g_q and $\hat{s}_q$ and hence we get $g_q = \frac{1}{n}\left[\sum \hat{I}(z) - \hat{s}_q \sum z\right]$ and $\hat{s}_q = \frac{n\sum z \times \hat{I}(z) - \sum z \sum \hat{I}(z)}{n\sum z^2 - (\sum z)^2}$ where, n is the number of pixels in each block.

3.4 Gradient Method for Domain Range Blocks

In the process of finding domain blocks for each range block, the practice of searching is to take a norm for the difference of block matrices, which is done iteratively pixel-by-pixel, as a result, NP time complexity exists. By searching only the central pixel based on gradient weight instead of pixel-by-pixel, NP can be reduced to P-class time complexity, and this is done using the norm of the central pixels' difference of the blocks. Once for each range block, there is the best-fit domain block, *DQ,* then there exists a set of IFS, $w_q(DQ)$ and for every $w_q(DQ)$, a 5-tuple is constructed $[(i, j), (t_q), \hat{s}_q, g_q]$, which represent the position, nature of transformation, brightness and contrast of the selected block accordingly. This method can be illustrated through the following example: Let the 3×3 block be defined as:

$$\begin{bmatrix} p_1\ p_2\ p_3 \\ p_4\ p_5\ p_6 \\ p_7\ p_8\ p_9 \end{bmatrix}$$

where:

- p_5 is the central pixel
- The others p_1 to p_9 are its 8-neighbours

A. Gradient Operators (Sobel-Based Approximation):
Using Sobel filters, which are commonly used to estimate gradients:

- Horizontal Gradient G_x:

$$G_x = (p_3 + 2p_6 + p_9) - (p_1 + 2p_4 + p_7)$$

- Vertical Gradient G_y:

$$G_y = (p_7 + 2p_8 + p_9) - (p_1 + 2p_2 + p_3)$$

B. Gradient Magnitude G:
To quantify the overall edge strength at the center:

$$G = \sqrt{G_x^2 + G_y^2}$$

or for computational efficiency:

$$G \approx |G_x| + \left|G_y\right|$$

Matching process:

- Compute g for both domain and range 3×3 blocks
- Compare the gradient magnitudes or use them as weights for the matching score
- This approach captures edge direction and texture strength, improving match accuracy over intensity-only matching.

The final matching score is computed by combining the weighted contributions of the central pixel and its neighbours. It is calculated as follows:

$$\text{Matching Score} = 0.6 \times \left(\frac{|G_D - G_R|}{G_D}\right) + 0.4 \times (\text{Neighbor Gradients Differences})$$

This score will be lower if the domain and range blocks are more similar in terms of their gradient profiles, leading to a better match.

In this example, the central pixel and its neighbors are compared based on gradient values, giving more weight to pixels with higher gradient magnitudes. This allows the matching process to take into account not just the pixel intensity but the change in intensity across the block, making it more robust to noise or slight variations in pixel values.

3.5 Low-Gradient Image

In flat or low-texture image areas, where gradients go to vanish or give little information, a hybrid matching strategy is used. In particular, such as matching based on mean intensity difference or local variance, if the average gradient magnitude g in a 3×3 window is below a threshold τ. The following values can be defined.

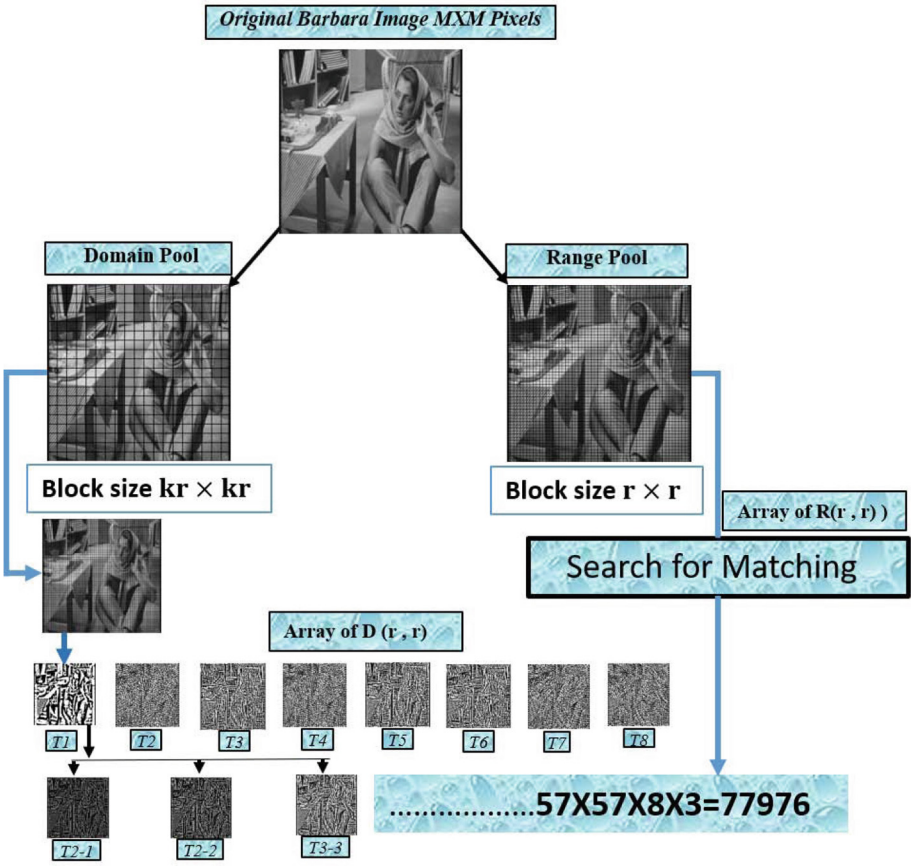

Fig. 1. Image blocking and 8 isometric transformed images, each has 3 scale factors to adjust brightness

- $\tau \approx 10$ for 8-bit images.
- The Euclidean distance between the mean values of range and domain blocks is used as a similarity metric

Picking the central pixels of both domain and range blocks, we find the best-fit domain block for each range block depending on less residual error between the two central pixels. Please see Figs. 1 and 2.

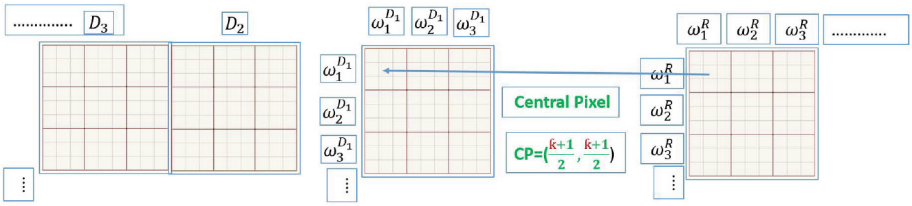

Fig. 2. Central pixels of range and domain windows

3.6 Encoding Process Flow

To encode a statistically self-similar original image, $I_M = [p_V \times p_H \mid p_V = p_H]$, we need to make two copies of the sets. The set of Domain $I_{M_D}(p_V, p_H)$ and Range $I_{M_R}(p_V, p_H)$. We split $I_{M_R}(p_V, p_H)$ into the uniform size of block partitions comprising nine pixels in each. The eight neighboring pixels of the block are symmetric to the central pixel in Fig. 4. For the domain set, $I_{M_D}(p_V, p_H)$, we repeat the process, taking $[\kappa 3 \times \kappa 3]$ pixels in each. The ultimate target of the encoding process is to generate the IFS parameter $\left[(i, j), \left(t_q\right), \hat{s}_q, g_q\right]$.

3.7 Decoding Process Flow

In the existing image decoding methods, pixel-by-pixel error checking is used, but in this study, we proposed a window-by-window approach to minimize the time.

1. The parameters from $T_i(\alpha, \beta) = [(i, j), (t_q), \hat{s}_q, g_q]$ are used to generate the IFS in the encoding algorithm.
2. To regenerate the range blocks, we used any arbitrary image with the same size as the original image.
3. Each range block is filed according to its location in T.
4. Step 3 is applied iteratively until the high-quality image is generated, as shown in Fig. 5.

The flow of the process is explained in Fig. 6.

4 Experiment, Results, and Discussion

In Tables 1, 2 and 3, we showed two statistical parameters, MSE and PSNR, which can tell the objective quality of the five images. In Table 1, we showed existing method results, and in Table 2 proposed the method; in Table 3, we showed the comparison between the methods. Table 2: full fractal encoding process time: 2×2 average pooling vs 2×2 max pooling when image size equals 256×256 and block size: 8×8 pixels.

The MSE and PSNR of the original and decoded images are calculated. We got influential and motivated results in the proposed method compared to an existing method in encoding and decoding improvement of 34.34% and 39.66%, respectively, for the same images. Moreover, we analyzed the gray scale for the same image and got a high class of improvement in encoding and decoding, 46.32%, respectively shown in Table 2.

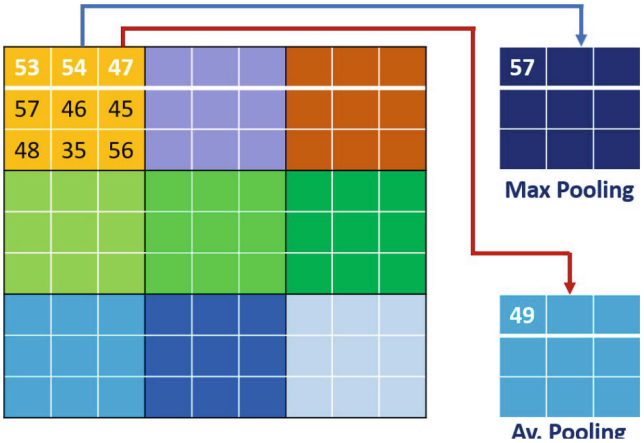

Fig. 3. Pooling system: 3 × 3 max and average pooling

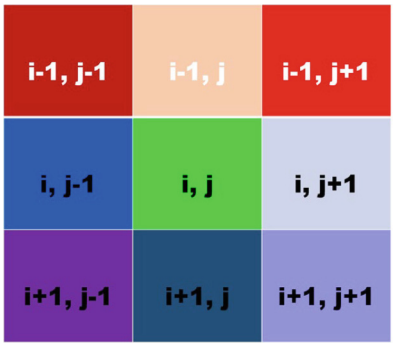

Fig. 4. The 8 neighbouring pixels of a 3 × 3 block



(a) (b) (a) (b) (a) (b) (a) (b) (a) (b)

Fig. 5. The proposed method: (a) Original standard images, (b) the corresponding decoded images.

FSIM: *Feature Similarity Index*. It is used to measure the sensitivity to edges and structure.

GMSD: *Gradient Magnitude Similarity Deviation*. It is used to measure the reflect gradient consistency.

PSNR (dB): *Peak Signal-to-Noise Ratio*. It is used to measure the quality of the reconstruction image.

Table 1. Encoding and Decoding Time Analysis

Image Name	Encoding Time(s)				
	M(2 ×2)	M(3 ×3)	TTR	I% TR	Av% TR
Lena	118.69	76.64	35.4	30.78	
Barbara	120.8	75.53	42.1	36.61	
Cameraman	121.57	75.08	36	33.03	34.34
Peppers	121	75.38	46	39.66	
Einstein	121.03	75.396	37	31.62	

Table 2. The fractal encoding process time.

Image Name	2 × 2 Max-pooling		2 × 2 Av-pooling	
	Time	PSNR	Time	PSNR
Barbara	54.7310	24.2654	74.1820	43.4059
Camera-Men	31.9190	25.0289	36.9030	45.6515
Einstein	33.3570	25.6204	57.7310	44.9906
Lena	33.5450	32.4749	59.3550	54.3088
Peppers	49.5590	26.6530	67.4640	47.1608

Mape: *Mean Absolute Percentage Error*: It is a measure used to find the average percentage difference between the reconstructed and original images.

Table 5 confirms that the very low or the very high central weights are assigned to the central pixel of one of the chosen images, the range between 0.5 and 0.7 consistently produces optimal PSNR values and faster encoding. This result validates our initial selection of $wc = 0.6$.

In some cases, when the average gradient magnitude is less than a specific threshold, we use some traditional features such as pixel mean or variance as intensity-based matching, as you can see in Table 6. This hybrid approach preserves performance and stability even in the presence of weak gradient information.

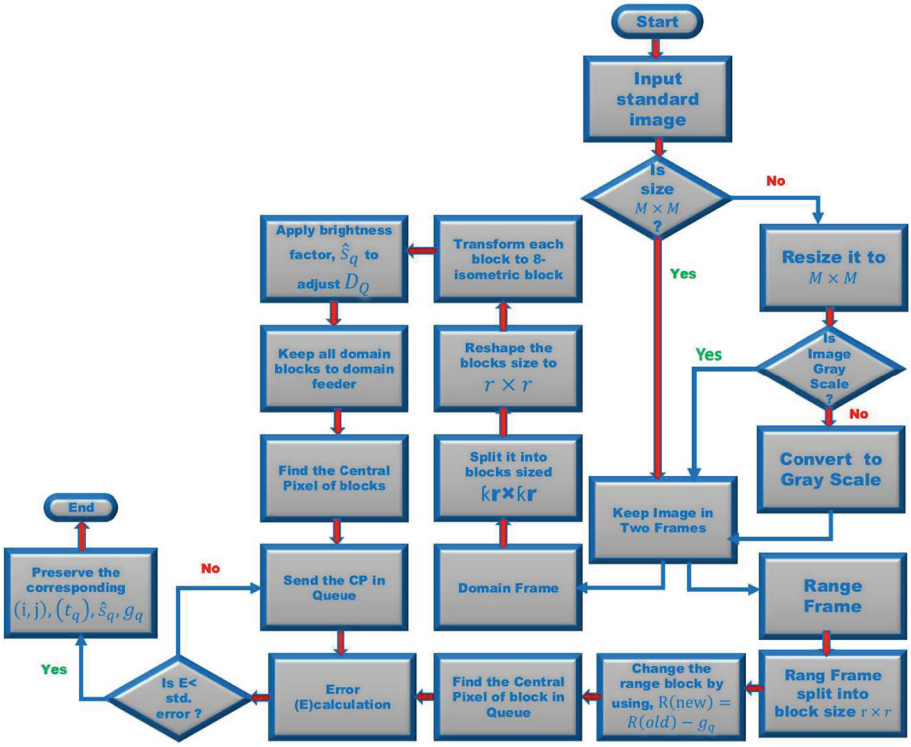

Fig. 6. Encoding process flow diagram

Table 3. Image quality index for decoded image analysis by full reference

IQAT	Barbara	Cman	Einstein	Lena	Pepper
MSE	40.3008	17.2353	28.6325	12.8956	24.7371
GMSD [34]	0.1952	0.1567	0.173	0.1422	0.1758
Rvalue:	0.8245	0.8745	0.8487	0.9231	0.8494
RMSE:	6.6793	5.4604	6.1975	4.4278	6.1738
FSIM [33]	0.8388	0.8615	0.853	0.9017	0.8522
Mape:	23.6426	11.0176	8.4698	4.3242	8.984

Table 4. Pooling Stage: Time and Image Quality Comparison with 3 × 3 average vs max pooling when Image Size: 252 × 252 pixels.

Image Name	3 × 3 Max-pooling		3 × 3 Av-pooling	
	Time	PSNR	Time	PSNR
Barbara	0.2190	22.2382	0.3280	41.6035
Camera-Men	0.2030	21.4793	0.3590	40.5847
Einstein	0.1720	23.2147	0.3750	43.0182
Lena	0.1880	27.2500	0.3750	48.9911
Peppers	0.2340	23.1677	0.4530	43.2268

Table 5. Different weight values

Central Weight (wc)	PSNR (avg)	Encoding Time (avg)
0.3	25.1 dB	55.3 s
0.4	26.2 dB	54.1 s
0.5	26.8 dB	52.7 s
0.6	**27.3 dB**	**51.2 s**
0.7	27.1 dB	52.0 s

Table 6. Performance comparison of flat and low-texture images

Image	PSNR (Grad-Only)	PSNR (Hybrid)	SSIM (Grad-Only)	SSIM (Hybrid)
Lena	27.2	**32.4**	0.89	**0.95**
Einstein	25.6	**30.1**	0.88	**0.94**

5 Conclusion

Fractal image encoding and decoding are considered important techniques because of their high compression ratios and resolution independence at any scale. Encoding and decoding are the two main stages of this technique. The process of encoding creates numerical data from the statistically self-similar input image called an Iterated Function System (IFS). Whereas, the decoding process uses the IFS data to reconstruct the original fractal image. This technique has a long encoding time due to the exhaustive block searches in partitioned iterative function systems, which reduces its use and is considered a challenge. One of the great improvements in encoding and decoding time is the use of 3 × 3 method. Gradient values are used to compare the central pixel and its neighbours, giving pixels with larger gradient magnitudes more weight. This makes the matching process more resilient to noise or minute changes in pixel values by allowing it to consider not only the pixel intensity but also the change in intensity throughout the

block. We believe the most important feature of Fractal Decoding that we proposed in this paper attained the high quality of the image even when zooming out the decoded image. Furthermore, due to its efficiency, this approach for image coding can be applied in Medical Imaging, where we need to focus on image details with quality, and in Surveillance Systems, when trying to get a clear picture of the intruder or the cause of the alarm. From the obtained results and through analysis, we can conclude that the proposed method has a clear advantage over the 2×2 Homogeneous Contraction method. The results we presented are obtained using MATLAB Software. More adaptive block size techniques, such as entropy-based segmentation and quadtree partitioning, could be used to improve the speed and quality of encoding. Combining such adaptive strategies with our gradient-based matching is a future direction of work.

Acknowledgements. This research was sponsored by the Universiti Malaya Research Excellent Grant UMREG 068 - 2024 (UM.0000285/HRU.RC), and the Universiti Malaya Research Maintenance Fee RMFI2067-2021 (UM.0002145/HIP.IP).

Author Contributions. All authors have contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.

Funding. UM.0000285/HRU.RC and UM.0002145/HIP.IP.

Conflict of Interest. All authors declare no conflicts of interest in this paper.

References

1. Duh, D.-J., Jeng, J.-H., Chen, S.-Y.: DCT-based simple classification scheme for fractal image compression. Image Vis. Comput. **23**(13), 1115–1121 (2005)
2. Davoine, F., Antonini, M., Chassery, J.M., Barlaud, M.: Fractal image compression based on Delaunay triangulation and vector quantization. IEEE Trans. Image Process. **5**(2), 338–346 (1996)
3. Koli, N.A., Ali, M.S.: Lossy color image compression technique using fractal coding with different size of range and domain blocks. In: International Conference on Advanced Computing and Communications, ADCOM 2006, pp. 236–239. IEEE (2006)
4. Jacquin, A.E.: Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Process. **1**(1), 18–30 (1992)
5. Mandelbrot, B.B.: Fractal geometry: what is it, and what does it do? Proc. R. Soc. Lond. A Math. Phys. Sci. **423**(1864), 3–16 (1989)
6. Barnsley, M.F., Massopust, P., Strickland, H., Sloan, A.D.: Fractal modeling of biological structures. Ann. N. Y. Acad. Sci. **504**(1), 179–194 (1987)
7. Ibrahim, R.W., Yahya, H., Mohammed, A.J., Al-Saidi, N.M., Baleanu, D.: Mathematical design enhancing medical images formulated by a fractal flame operator (2022)
8. Al-Saidi, N.M., Abdul-Wahed, H.Y.: Classification of remote sensing images via fractal discriptores. In: 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), pp. 99–104. IEEE, March 2018
9. Al-Saidi, N.M., Said, M.R.M.: Biometric identification using local iterated function. Eur. Phys. J. Spec. Top. **223**(8), 1647–1662 (2014). IEEE transactions on Image Processing, 20(8):2378–2386, 2011

10. Grinchuk, P.S., Danilova-Tretiak, S.M.: Fractal power law and polymer-like behavior for the metro growth in megacities. Chaos Solitons Fractals **194**, 116137 (2025)
11. Barnsley, M.F.: Fractals Everywhere. Academic Press, Cambridge (2014)
12. Hutchinson, J.E.: Fractals and self-similarity. Indiana Univ. Math. J. **30**(5), 713–747 (1981)
13. Kominek, J.: Algorithm for fast fractal image compression. In: Digital Video Compression: Algorithms and Technologies 1995, vol. 2419, pp. 296–306. International Society for Optics and Photonics (1995)
14. Al-Saidi, N.M., Al-Bundi, S.S., Al-Jawari, N.J.: A hybrid of fractal image coding and fractal dimension for an efficient retrieval method. Comput. Appl. Math. **37**(2), 996–1011 (2018)
15. Al-Azawi, R.J., Al-Saidi, N., Jalab, H.A., Ibrahim, R., Baleanu, D.: Image splicing detection based on texture features with fractal entropy (2021)
16. Al-Saidi, N.M., Md Said, M.R.: Improved digital signature protocol using iterated function systems. Int. J. Comput. Math. **88**(17), 3613–3625 (2011)
17. Tseng, C.-C., Hsieh, J.-G., Jeng, J.-H.: Fractal image compression using visual-based particle swarm optimization. Image Vis. Comput. **26**(8), 1154–1162 (2008)
18. Zhu, L., Zeng, X., Chen, B., Chen, P., Li, Y.H., Wang, S.: Leveraging diffusion knowledge for generative image compression with fractal frequency-aware band learning. arXiv preprint arXiv:2503.11321 (2025)
19. Lee, C.K., Lee, W.K.: Fast fractal image block coding based on local variances. IEEE Trans. Image Process. **7**(6), 888–891 (1998)
20. Wang, Q., Du, J.: Modified decoded image quality prediction method for fractal image coding. Comput. Electr. Eng. **118**, 109462 (2024)
21. Ahadullah, M., Sapar, S.H., Al-Saidi, N.M., Said, M.R.M.: Competitive improvement of the time complexity to encode fractal image: by applying symmetric central pixel of the block. IEEE Access **9**, 5028–5045 (2020)
22. Long, B., et al.: Improved fractal coding and hyperchaotic system for lossless image compression and encryption. Nonlinear Dyn., 1–30 (2024)
23. Al-Saidi, N.M.G., Ali, A.H.: Towards enhancing of fractal image compression performance via block complexity. In: 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), pp. 246–251. IEEE (2017)
24. Galabov, M.: Fractal image compression. In: Proceedings of International Conference on Computer Systems and Technologies (CompSysTech 2003), pp. 320–326 (2003)
25. Datta, A., Parui, S.K.: A robust parallel thinning algorithm for binary images. Pattern Recognit. **27**(9), 1181–1192 (1994)
26. Fisher, Y.: Fractal Image Compression: Theory and Application. Springer, New York (2012). https://doi.org/10.1007/978-1-4612-2472-3
27. Xing, C., Ren, Y., Li, X.: A hierarchical classification matching scheme for fractal image compression. In: 2008 Congress on Image and Signal Processing, vol. 1, pp. 283–286. IEEE (2008)
28. Caso, G., Obrador, P., Jay Kuo, C.-C.: Fast methods for fractal image encoding. In: Visual Communications and Image Processing 1995, vol. 2501, pp. 583–595. International Society for Optics and Photonics (1995)
29. Fisher, Y., Menlove, S.: Fractal encoding with HV partitions. In: Fisher, Y. (ed.) Fractal Image Compression, pp. 119–136. Springer, New York (1995). https://doi.org/10.1007/978-1-4612-2472-3_6
30. Hebert, D.J., Soundararajan, E.: Fast fractal image compression with triangular multiresolution block matching. In: International Conference on Image Processing, 1998. ICIP 98. Proceedings, vol. 1, pp. 747–750. IEEE (1998)
31. Han, J.: Fast fractal image compression using fuzzy classification. In: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, pp. 272–276. IEEE (2008)

32. Al-Bundi, S.S., Al-Saidi, N.M., Al-Jawari, N.J.: Crowding optimization method to improve fractal image compressions based iterated function. Int. J. Adv. Comput. Sci. Appl. **7**(7) (2016)
33. Xue, W., Zhang, L., Mou, X., Bovik, A.C.: Gradient magnitude similarity deviation: a highly efficient perceptual image quality index. IEEE Trans. Image Process. **23**(2), 684–695 (2014)
34. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. IEEE Trans. Image Process. **20**(8), 2378–2386 (2011)