IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 6, 2020, accepted November 29, 2020, date of publication December 14, 2020, date of current version January 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3044290

Competitive Improvement of the Time Complexity to Encode Fractal Image: By Applying Symmetric Central Pixel of the Block

MD AHADULLAH[1], (Member, IEEE), SITI HASANA SAPAR[1,3], NADIA M. G. AL-SAIDI[2], (Member, IEEE), AND MOHAMAD RUSHDAN MD SAID[1]

[1]Institute for Mathematical Research, University Putra Malaysia, Serdang 43400, Malaysia
[2]Faculty of Science, University of Technology, Baghdad 12906, Iraq
[3]Faculty of Science, University Putra Malaysia, Seri Kembangan 43400, Malaysia

Corresponding author: Md Ahadullah (md.ahad1234@gmail.com)

This work was supported by a research grant from the Ministry of Education, Malaysia, under vote number: Putra Grant UPM/7002/1/GBP/2017/9597900.

ABSTRACT By combining the basics of self-similarity, scaling correlation, and statistical components, Benoit Mandelbrot formulated the idea of a natural fractal entity, an entity described by those fundamentals. As a result of these principles, fractal image codings are being used in many substantial applications already, such as image compression, image signature, image watermarking, image attribute extraction, and even image texture segmentation. Thus, while fractal image coding is relatively new in the field of image encoding, it has gained broad acceptance at a rapid pace. In light of its beneficial qualities, such as quick decomposition, high compression ratio, and the independence of resolution at any size make these applications conceivable. However, compared to its advantages, fractal image coding is extremely time-complex and so remarkably expensive, which hinders its prevalence. A wide hunting domain blocks for the relevant range blocks caused this difficulty. We proposed several improvements to the Jacquin design in this paper. We first used max-pooling as an alternative for the medium bonding of spatial contractions to validate the value of the edge textures of the block. Secondly, we construct the odd-size pixel block alternative to an even-size pixel block for validation of the symmetric central pixel (CP). Finally, before the search started, we proposed a shortening of block space, using the central pixel of the block to convert each eight-bit pixel to a two-bit pixel. As a consequence, the symmetrical CP of odd pixels block, reduction of block space, and edge pixel selection accomplished faster coding and competitive image quality than existing known exhaustive search algorithms.

INDEX TERMS Central pixel, statistical self similarity, symmetrical central pixel, IFS-Data.

I. INTRODUCTION

Fractal geometry, according to [65], can emulate natural objects in a fake perspective rather than a real one. It was not possible to decode the image of a natural object before discovering the fractal geometry. Mandelbrot combined the Julia set in a single pattern by continuous iteration of a complex function until finally reaching an attractor and conceptualized three substantial ideas in a single frame here, first, the constitution of self-similarity of a natural object; second, the scaling dependence that opened the new door of dimensionality as a fractional dimension; finally, the statistical features of natural objects ([1]). Immediately based on Mandelbrot's research, [45] presented his significant mathematical contribution concerning the existence and uniqueness of a attractor as well as iterative operators of contraction. Following this development, [5] claimed the reproduction of natural image as a fractal attractor by introducing Hutchinson's iterative operators. Barnsley named this process the Iterative Function System (IFS). An IFS coded image

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

constitutes of two steps: First, fractal image encoding that generates IFS, and Second, fractal image decoding uses its collage theorem to reproduce almost even attractor based on a set of the best IFS fit. Nevertheless, Barnsley's claim has not been successfully applied in the reproduction of the original image in automated self-regulating fashion since natural objects are not strictly self-similar, but are partially or statistically self-similar. Following this, [48] proposed the Partition Iterative Function System (PIFS), becoming the first to claim that the similarity of natural objects could be parts to parts and not always wholes to parts. This research elicited a quick response in the image processing community. However, in Jacquin's method, the encoding time complexity was massive due to a large number of domain blocks searching for each range block. Thus, researchers began to focus on how to reduce encoding time. A panel of researchers attempted to reduce the encoding time by using elementary methods such as clustering domain blocks, classifying blocks by choosing geometrical or statistical features, and applying the Pixel approximation, Block partitioning, Hybrid technique, Adaptive search method, Entropy search method, Block size modification, Heuristic technique, Tree structure, Parallel implementation, Features based technique, No search method. However, following the Jacquin's Technique, [6] changed the search method for codebook to generate address data that displayed a Gaussian-like distribution and coded more effectively than a uniform (random) distribution by using an entropy coding method. Reference [6] took a 12 × 12 pixels domain block while range block size is 4 × 4 pixels. so every domain block consists of nine range blocks. Reference [72] applied the inner product to confirm the error metric of the library and target blocks. Even [49] showed classifications of blocks based on block geometry. Reference [35] classified the domain blocks into 72 categories with combinations of mean and variance and searched class wise instead of all. Reference [71] used triangular form blocks identified by its triplet of co-ordinate points where the matching blocks theory would have been the typical angle points.

In quadtree partitioning, [64] implemented variable range blocks by describing the reduction bit allocation as 0.29. Reference [53] presented a coder that translated a block of pixels into a position vector, for example, a 4 × 4 range block becomes a 16-dimensional position vector. Consequently, the coder takes the mean and variance of those position vectors, which works as the identity of the blocks and uses b-tree for indexing and finds the nearest domain blocks. Reference [73] first proposed to convert the gray valued pixels in blocks into a binary pattern locally in texture classification. References [85] and [102] omitted certain domains based on a few basic standards. Reference [90] portrays an enhanced formulation on the approximation of the nearest neighbor search grounded on orthogonal projection and pre-quantization of the fractal transmute parameters. Besides, to further boost the efficiency of the new algorithm, an optimized adaptive scheme for the estimated search parameter is extracted. Reference [74] transformed eight neighboring pixels into a binary arrangement with clockwise rotation shown in Figure 2. Then, the patterns with CP formed a particular block's identity, for more detail ([76], [77], [86]).

On the other hand, we suggested transforming all pixels in the domain and range blocks into binary patterns at the final stage of this analysis. After that, to get the optimum distance between them, we applied the Hausdorff Metric to the binary value of the blocks. The overall works in this paper are to modify a series of stairs, the size of the block, the spatial contraction factor, and its mechanism and ultimately usages of the Central Pixel (CP) that converts block-wise each 8-bits pixel into a 2-bits. Integrating all modifications in a single frame, we tested a set of 36 standard commonly used images shown in Figure 4 and displayed the encode time of several sizes of images and blocks shown in Table 3, while Figures 15, and 16 display visual presentations. Besides, we considered several Image Quality Evaluation Techniques (IQETs) to explain image quality and obtained a series of good results. In Max Pooling Odd Block Pixel of Binarization using Central Pixel (MPOBPBCP) method, we have achieved a dramatic result of about 89.82% encoding time improvement on average compared to several known existing methods shown in the Table 8 with competitive image quality displayed in Table 5.

The paper is arranged as follows: in section II, we focused on literature review, in section III, we showed theoretical considerations; in section IV, we included proposed modification and its methodology on fractal coding; in Section V, we present proposed algorithm of the method, MPOBPBCPM; in Section VI, we kept detailing of the experimental results; in section VII, we present conclusions and further directions of improvement.

II. LITERATURE REVIEW AND ANALYSIS

After years of research, several Fractal Image Compression (FIC) and other General Image Processing Methods (GIMP) as well as Image Quality Evaluation Techniques (IQET) developed and we demonstrated in Table 1. It is important to discuss and review this literature.

A. FRACTAL IMAGE COMPRESSION (FIC)

We present a few approaches of FIC that has been developed into the current stage based on the input parameter and expected output.

1) MODIFICATION OF BLOCK DIMENSION

Reference [15] proposed the $B \times B$ and $(3B - 1) \times (3B - 1)$ range and domain block sizes, claiming that the encoding time decreased significantly when the finite-state vector quantization (VQ) approach combined with partition base fractal image coding with some adjustment and finally, encoded image with 0.19 bpp. For each $n \times n$ range block, [21] suggested the domain pool as $3n \times 3n$, $4n \times 4n$ $5n \times 5n$ or $6n \times 6n$, reducing the computational time needed for a brute force search to optimize image quality. Conci applied the local fractal dimension (FD) to subdivide the blocks into four classes. Reference [52] research on variable sizes of

TABLE 1. **Tabular format of literature review: A. Fractal Image Compression (FIC): 1) Modification of Block Dimension (MBD), 2) Average Pooling Method and Pooling Factor (APMPF) 3) Block Partitioning (BP) 4) Block classification (BC) 5) Reduction of ISO-Metrics Mapping in numbers (R-ISO-MM) 6) Tree Structure (TS) 7) Features Based Technique (FBT) 8) Parallel Implementation (PI) 9) No search Method (NSM). B. General Image Processing Methods (GIPMS): 1) The CP and Block Dimension (CPBD) 2) Spatial Contraction Method (SCM) 3) Block Binarization Using Central Pixel (BBUCP). C. Image Quality Evaluation Techniques (IQETs): 1) Objective Image Quality Evaluation Techniques (OIQET) 2) Bits Allocation Techniques (BAT).**

MAOLR	SAOLR	Several Known Methods				
A. FIC	1) MBD	[15]	[21]	[52]	[20]	
	2) APMPF	[46]	[6]	[114]	[62]	[3]
	3) BP	[72]	[67] and	[87]	[101]	
	4) BC	[103] [8]	[92] [107]	[20] [70]	[105] [78]	[95] [55]
	5) R-ISO-MM	[104]				
	6) TS	[4]	[4]	[38]		
	7) FBT	[89]	[39]	[93]	[62]	
	8) PI	[91]	[75]			
	9) NSM	[36]				
B. GIPMS	1) CPBD	[40]	[74]	[63]	[42]	[94]
	2) SCM	[11]	[51]	[112]		
	3) BBUCP	[73] [31]	[74] [32]	[77] and [32]	[86] [100]	[76] [19]
C. IQETs	1) OIQET	[97] [115] [61]	[110] [17] [30]	[109] [108]	[111] [108]	[16] [30]
	2) BAT	[25] [9]	[24]	[89]	[22]	[27]

range and domain blocks in the fractal image coding to know the effect of variability. Reference [20] explain a method of dividing the original image into homogeneous blocks and achieving a significant speedup compared to the exhaustive search method.

2) AVERAGE POOLING METHOD AND POOLING FACTOR

The average pooling layer in Convolutional Neural Networks records the average pixel values in the template. ([3]). It results in smoothing the pixel intensity of the block by averaging entire pixels in block shown in Figure 9. In Fractal Image compression, [46] used it with pooling factor of $\frac{1}{2}$. For the same work, [6] takes the spatial contraction factor of $\frac{1}{3}$ with even size of pixels block. Reference [114] used Lena, Peppers, F16 and Boats images of size 256 × 256 pixels with contraction Factor of $\frac{1}{2}$. Reference [62] uses Lena image of size 256 × 256 pixels, range 8 × 8, pooling factor $\frac{1}{2}$.

3) BLOCK PARTITIONING

Reference [72] proposed fixed square block partition (FSBP). After that, [67] and [87] also use FSBP. These researches show that FSBP is the simplest of all square block partitioning schemes. In addition, [101] states that this type of block partition is efficient in transforming individual object block coding.

4) BLOCK CLASSIFICATION

Reference [34] classified the domain blocks into 72 categories and then searched by group. Reference [84] performed on empirical studies using Fisher's adaptive quadtree method by excluding all unlikely domains and claimed a greater speed by restricting the search to a reduced domain pool where Soupe employed a parameter $\alpha \in (0, 1]$. A merging scheme with quadtree segmentation was proposed by [18] that would combine many analogous surrounding blocks into a region. Chang states that each block is initially a region, and fuse them geometrically if each area pair is adjacent. Chang then determines the coefficients of the iteration function method and the corresponding error for the region pair where the coder can correctly merge the region pair with the lowest error, and the merging process continues until the error reaches the selected threshold. Reference [20] modified Fisher's quadtree approach, what is quadrant decomposition depending on the threshold value for the best match. Stating the drawback of it, Chung proposed a sub-block approach to partition the whole of original image into a set of variable-size blocks with an interpolation technique and got an improved result. Chung set the threshold values of the collage error, and ε of 1500 and 4, respectively.

Reference [103] uses block variance and mean to classify image blocks, and so the range and domain blocks get their respective classes based on the mean value, and thus reduces the domain blocks searched for each range block. Wu contended that the proposed algorithm achieved the same reconstructed image quality as the exhaustive search and would considerably reduce the run-time required.

Reference [92] utilized the neighboring concept used to the improved vector quantization (VQ) method and named it as a spatial correlation. Truong denoted four types of range blocks as vertical, horizontal, and two diagonals and matched domain to range blocks with the corresponding direction by using MSE provided a threshold. So, it is full search with classified domain and range blocks. Truong's and Jacquin's Methods are almost similar except the classification of range blocks. Reference [105] improves Fisher's layout and achieved 576 classes by using mean pixel value and variance of blocks.

Reference [95] introduced a new two-step FIC acceleration algorithm. First, the condensed statistical vector expressions will speed up encoding twice as fast as the Baseline fractal compression (BFC) without the BFC resulting loss of image quality. Second, based on the fact that affine self-similarity is equal to the absolute value of Pearson's coefficient of correlation, a new block classification approach of compact arrangement sets is introduced to speed up coding process. Results of the experiment and theoretical study demonstrate that the suggested scheme achieves high performance in both maintaining image quality and encoding efficiency.

Reference [8] suggests a sub-image classification system that hierarchically partitions the domain pool and compares a range to only those domains that belong to the same hierarchical group as the range It is a Fisher's 24 model extension research. Reference [107] uses a primal-dual algorithm along with a median-based Fisher classification scheme to accelerate the encoding speed and thus achieves the robust fractal image coding. Reference [70] proposed a modified hierarchical classification strategy for fractal image coding using adaptive quadtree partitioning to improve the compression ratio of the lossless coding scheme.

Reference [78] presents a new way of mapping the domain and the range blocks based on peer adjacent and mean difference methods in which groups of four blocks share their union as a common peer-adjacent domain block. Only range blocks, R. in that domain (second hierarchy) are searched for a given domain block, D in the top Hierarchy. Each range in the second hierarchy will become a lower-hierarchy domain and will continue to map ranges in the same domain. This can reduce the number of computations, resulting in a much faster runtime. Smaller range size provides more computational time because more matching processes need to be calculated, thus saving only small storage space.

Reference [55] introduces an approach to estimate fractal texture identification's affine parameters to reduce computational complexity by splitting image into several blocks of various sizes. They used a variety of data chunks such as Fractal Pattern Chunk (FPC), Intermediate Raster Chunk (IRC) and more. Using these chunks, an image is stored in compressed form.

5) REDUCTION OF ISO-METRICS MAPPING IN NUMBERS

Reference [104] proposes a method by utilizing the best polynomial approximation to decide whether a block of a domain is fairly similar to the block of specified range. They use just 2 or 4 isometrics to speed up compression in the fractal code instead of 8 isometrics claiming that the probability distribution of 8 isometrics is not average.

6) TREE STRUCTURE

Reference [4] obtains quantitative results about the measure of distance employed in the search by deriving an incremental procedure for bounding the pixels in the domain row. In a tree structure, [4] organizes the domain blocks, and uses the scheme to guide the search. Reference [38] introduces the number and locations of the local extreme points in a row inside an image block known as the classification features composed of a three-layer tree classifier for the search distance of similarity matching to speed the encoding cycle preserving the accuracy of the reconstructed image.

7) FEATURES BASED TECHNIQUE

References [56], [57] suggested an effective method of zero contrast prediction to determine whether or not the contrast factor for a domain block is zero, and to measure the corresponding difference between the range block and the transformed domain block efficiently and precisely. Reference [29] proposed a classification scheme of three classes based on the edge properties of image blocks in the sense of self-similarity and claimed improvement using a class-based threshold. Reference [89] differentiated the blocks by using standard deviations. Reference [39] applies a fuzzy structure cataloger to sort image blocks and on the other hand, for the same purpose [93] serves Particle Swarm Optimization. Reference [62] use the fractal encoding approach for statistical loss analysis and construct a box-plot to identify the loss value distribution, then split it into several sections, and assign it to the specified model. Eventually, experimental results show the method's efficacy.

8) PARALLEL IMPLEMENTATION

Reference [91] states that the dihedral operations for assigning domain blocks require eight separate MSE computations between the specified range block and the chosen domain block, but these eight operations entail extensive computing in the frequency domain obtained by the use of the discrete cosine transform (DCT). Consequently, these DCT computations on a real-time processor are trivial, and make this new algorithm very realistic. Thus, parallel implementation is important for calculating the MSE using all the DCT coefficients of the eight orientations of a given image set. By converting into coordinates, Truong defines these eight orientations for a block of a domain. For each direction of the original block, Truong forms explicit DCT formulas. Using these formulas, the data under consideration can be grouped together and avoided the repeated calculations of the various operations and hence the computation time reduced. Given that the DCT has good energy compaction, only part of the DCT coefficients need to be used in realistic situations to find the scalar component. Thus, in the frequency domain, the coder will further reduce the computational difficulty in reducing least-square error. Reference [75] proposes the reduction of fractal encoder's time complexity, taking into account the data dependency between different operating modules, which effectively uses parallelism preserving the image quality.

9) NO SEARCH METHOD

Every time-consuming component [36] modifies is parameter search, error calculation, number of range blocks, and domain search for fractal coding and accelerated with new techniques as no search. Reference [36] splits the range block R into smaller sets and fixes the domain block position according to that of range block. By defining a concept as same degree, [36] tests the similarity between two blocks to induce the range block location and the corresponding domain block.

B. GENERAL IMAGE PROCESSING METHODS (GIPMS)

1) THE CP AND BLOCK DIMENSION

Reference [40] suggested Gray-Tone Spatial-Dependence Matrices and demonstrated on how resolution cells connected to the nearest neighbors in 3 × 3 and 4 × 4 pixels block (shown in Figure, 1). Reference [42] proposed a region descriptor named as center-symmetric local binary pattern (CS-LBP) descriptor which combines the strengths of the Scale Invariant Feature Transform (SIFT) descriptor [63] and LBP (Local Binary Pattern) [74] texture operator. Heikkila claimed the several advantages of the descriptor, such as tolerance to illumination changes, robustness on flat image areas, and computational efficiency. The Center Symmetric Local Binary Co-occurrence Pattern was suggested by [94], and the feature is extracted based on the diagonally symmetric elements around the center. Motivating on the concept of Center Symmetric and diagonally symmetric, we recommended odd size pixels for both blocks and spatial contraction size.

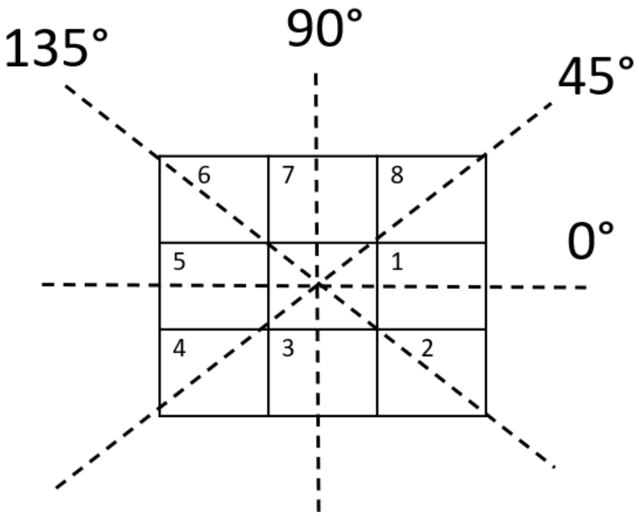
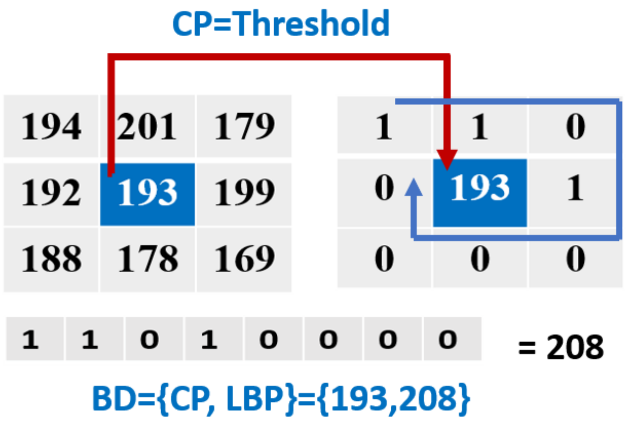


FIGURE 1. Gray-tone spatial-dependence matrices.



194	201	179
192	193	199
188	178	169

1	1	0
0	193	1
0	0	0

1	1	0	1	0	0	0	0

FIGURE 2. Block pixel binarizing model of [74].

2) SPATIAL CONTRACTION METHOD

Reference [11] claims that max-pooling as a Spatial Contraction has become popular recently because of its better performance and is suitable for sparse images. Reference [51] experiences that max-pooling works better than average pooling (not yet used in FIC) for computer vision tasks such as image classification. [112] displays both cases of even and odd pooling sizes based on the location information, turning it into displacement function and finds better results for max-pooling.

3) BLOCK BINARIZATION USING CENTRAL PIXEL

Reference [73] first proposed to convert the gray valued pixels in blocks into a binary pattern locally in texture classification. Reference [74] transformed eight neighboring pixels into binary patterns keeping the central pixel intact and finally made a pair between the central pixel and combining of other corresponding binary 8-neighboring pixels together, and thus, they formed a particular block's identity to recognize block in texture shown in Figure 2. References [77] and [76] use this idea in facial recognition. Reference [86] suggests an approach to image compression based on Local Binary Pattern (LBP) as a dispersion of local contrast by binarizing each neighbor of 3 × 3 matrix using the central pixel value as a threshold, resulting in an 8-bit binary code along with the initial value of the central pixel altogether a local block descriptor (BD) stored in a newly proposed Local Binary Compressed format. Szoke shows the results of improvement in compression ratio.

C. IMAGE QUALITY EVALUATION TECHNIQUES (IQETs)

There are two classes of IQETs, Subjective, and Objective. Besides, at the time of Fractal image encoding, bits allocation is also a vital part. We need to review Objective IQETs and bits allocation in this section.

1) OBJECTIVE IMAGE QUALITY EVALUATION TECHNIQUES

Reference [115] claims that the deterioration of decoded images in fractal coding derives mainly from the poor self-similarity of the input image induced by the collage errors in the encoding process, thereby producing a changed image that satisfies good self-similarity and well approximates the original image. Zou then suggested a procedure for optimizing the number of range blocks utilizing collage error. Reference [108] states that one can categorize IQETs as methods of full-reference (FR), no-reference (NR), and reduced-reference (RR). For example, the mean squared error (MSE), the peak signal-to-noise ratio (PSNR), and the Structural Similarity Index (SSIM) are FR methods. Reference [108] claims that No Reference SSIM (NSSIM) scores an image quality without previous training or learning and can automatically calculate image quality by measuring image luminance, contrast, structure, and blurriness. WPSNR is a reflection of PSNR metric generalization that calculates the perceptually weighted distorted parameter of a block using the Lagrange clockwise bit-allocation method in the encoder. WPSNR explains the visual benefit due to the results of two subjective standardized tests [30].

During the decoding process, encoded IFS-data extracted from the image was decoded into a fractal image by the decoder. Therefore, for every corresponding original image (reference), we need a group of numerical measurements of fractal image quality with the correct tools. Reference [32] grouped numeric measures of image quality according to their findings. However, we kept the Image Quality Evaluation Techniques (IQETs) into a single set comprising seventeen measures. These are as follows: The Mean Square Error (MSE) confirmed a pixel intensity error. An image signal test compared to its Noise is tested by SNR and PSNR. Further, Weighted Peak Signal to Noise Ratio (WPSNR) ([30], [61]) was used to validate PSNR. We applied the Structural Similarity Index Metric (SSIM) [100] to determine the similarity of structure. Besides, the Edge Strength Similarity Index Metric (ESSIM)([19]), and ([111]), Multi-Scale Structure Similarity Index (MSSSIM) [100], Spectral Similarity Index (SRSIM) [109], and No Reference Structural Similarity

Index Metric (NSSIM) [108] were used to certify SSIM values. To measure the similarity of feature information, we used the Features Similarity Index Metric (FSIM) [110]. We apply Contrast Per Pixel Variance (CPPD) [17] and Entropy Difference (ED) [97] to Contrast and Luminance tests. GMSD tool tested the Gradient Magnitude Similarity Deviation. Finally, the tools of Sparse Feature Fidelity (SFF) [16] and Image Fidelity Measure (IFM) ([31], and [32]) tested the fractal decoded image to justify the quality. In Table 5, we showed all numerical test results. We tried all of these seventeen quality tools to evaluate the decoded fractal image and got a set of competitive results.

2) BITS ALLOCATION TECHNIQUES (BATS)

During fractal image encoding, bits per pixel (bpp) is very significant. Original and fractal decoded image need to be confirmed the bits allocation. It varies on research needs, for example, Jacquin used 6 bpp in original image while Fisher used 8 bpp. It signifies the compression ratio. [22], [24], and [89] used 6 bits for each the domain blocks positioning in fractal image encoding. [22] used 3 bits for the eight isometrics. Reference [25] used 8 bits for the brightness adjust. [9] used 2 bits for the contrast adjust when quantize to 0.25, 0.50, 0.75, so bits required per block are 25 bits. As a result, the total bits required for a image to encode are $\frac{25\times ndb}{total\ pixel\ in\ image}$, and compression ratio, $CR = \frac{bpp_{uncomp}}{bpp_{comp}}$, ([27]). The way we approach requires 25 bits per block, so average bpp is 0.151.

D. GENERAL SUMMARY AND COMMENTS ON LITERATURE REVIEW

A fractal image encoding suffers from the optimal and competitive balance between encoding time and image quality in the spatial domain with even size pixel block and contraction factor, the key results from the above literature review. The possible explanations for this are the block dimensionality, contraction factors, contraction methods, time, and space required between two blocks for error findings in full-search. As a result, If we look into Table 9, it shows the average of PSNR, bpp, and ET(s) with respective image and block sizes of proposed methods. On the other hand, most recent research in the literature review shown in Tables 6, and 7 shows encoding time is a bit expensive compare to PSNR and bpp. All the results taken from the literature review are influential for future expansion to compress the images using fractal theory. The writers discussed a solution to the issues in Section IV.

III. THEORETICAL CONSIDERATIONS

In this section, we present the mathematical ideas behind fractal theory. For more details, see [2], [33], [37].

A. MATHEMATICS INSIDE FRACTAL THEORY

We take an image, I_M in complete metric space (I_M, f) comprising all the properties of metric space, where the set I_M is mapped to I_M again by a distance function f as

$$f : I_M \times I_M \rightarrow I_M. \tag{1}$$

Since I_M is an image vector so it has spatial coordinate and pixel intensity. If we introduce some other extra properties here to make it a Hausdorff space, we need to confirm that no open set in the space (I_M, f) has common elements. Let P and Q be two compact sets in I_M such that

$$P \cap Q = \Phi. \tag{2}$$

Hence if $p_x, q_y \in (I_M, f)$ then $\exists\, p_x \in P$ and $q_y \in Q$, $p_x \neq q_y$. The distance between the two sets P and Q in space (I_M, f) can be calculated by using the Hausdorff metric such as,

$$H(P, Q) = max\{h(P, Q), h(Q, P)\} \tag{3}$$

References [104], and [7] state that an object to be considered by IFS codes, three important steps are essential that can be written mathematically, Firstly, average Pooling, the process of smoothing is actually averaging of pixel intensity, $I_{x,y}$ block by block, where x and y are the pixel coordinates for square block and $I_{x,y} = f(x, y)$, and $\kappa \times \kappa$ average pooling. For details, reader can see [12], and [13]

$$T_p(I_{x,y}) = \frac{1}{\kappa}\sum_{x=1}^{\kappa}\left(\sum_{y=1}^{\kappa}\frac{1}{\kappa}(I_{x,y})\right). \tag{4}$$

Secondly, block intensity transformation, which can be found by using

$$T_I(I_{x,y}) = sI_{x,y} + o \tag{5}$$

where s and o are the contrast and brightness respectively. Finally, geometric transformation by using

$$T_g(I_{x,y}) = a_g I_{x,y} + b_g \tag{6}$$

where T_g is the transformation function and a_g and b_g are the coefficients of isometric transformation and translation vectors respectively, while $g = 1, 2, \ldots 8$, is a translation index. Thus, [7], [104] shows an overall transformation as follows:

$$T_{overall}(I_{x,y}) = T_p(I_{x,y}) \circ T_I(I_{x,y}) \circ T_g(I_{x,y}) \tag{7}$$

Now, if we apply this transformation $T_{overall}$ iteratively on all domain blocks separately, and combine them in a single plot, no doubt we can then reach a fixed point according to the Banach contraction mapping principle. This collection of all transformed domain blocks matched with the corresponding range blocks can be used to build a decoded image on the theory of collage. Mathematically,

$$I_{trans} = \bigcup_{n=1}^{N} T_{overall}(I_{x,y}). \tag{8}$$

Hence using Eq (3)

$$H(I_M, A) \leqq \frac{\varepsilon}{1-s} \tag{9}$$

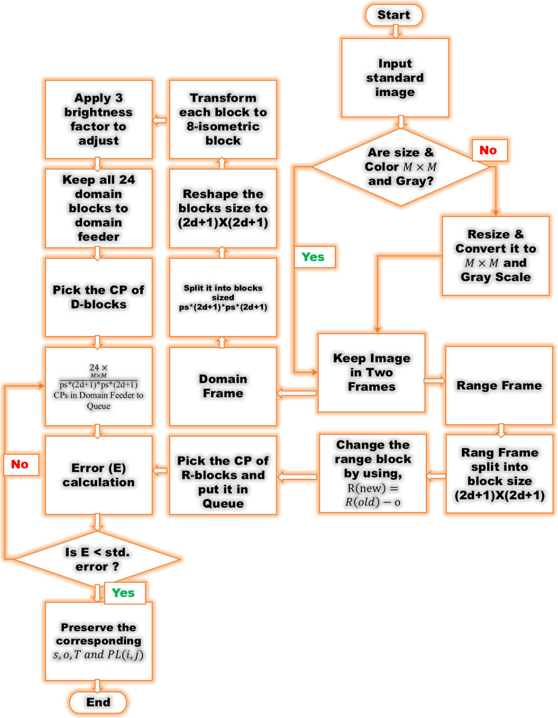

Start
Input standard image
Are size & Color $M \times M$ and Gray?
No
Resize & Convert it to $M \times M$ and Gray Scale
Yes
Keep Image in Two Frames
Range Frame
Rang Frame split into block size (2d+1)X(2d+1)
Change the range block by using, R(new) = R(old) − o
Pick the CP of R-blocks and put it in Queue
Domain Frame
Split it into blocks sized ps*(2d+1)*ps*(2d+1)
Reshape the blocks size to (2d+1)X(2d+1)
Transform each block to 8-isometric block
Apply 3 brightness factor to adjust
Keep all 24 domain blocks to domain feeder
Pick the CP of D-blocks
24 × M×M / ps*(2d+1)*ps*(2d+1) CPs in Domain Feeder to Queue
Error (E) calculation
Is E < std. error ?
No
Yes
Preserve the corresponding s, o, T and PL(i, j)
End

FIGURE 3. Encoding process diagram.

FIGURE 4. Original images Set-A.

FIGURE 5. Original images Set-B.

I_M is the original image and A is the fractal decoded image as an attractor in each instance. But in line with Lipchitz contraction mapping, we may have:

$$H(I_M, I_{trans}) \leqq \varepsilon. \tag{10}$$

So now, with the help of Eq (9) and Eq (10), we combine and establish the following:

$$H(I_M, A) \leq \frac{1}{1-s} H(I_M, I_{trans}). \tag{11}$$

Finally, after combining Eq (8) and Eq (11), we recognize

$$H(I_M, A) \leq (1-s)^{-1} H(I_M, \bigcup_{x=1}^{N} T_{overall}(I_{x,y})). \tag{12}$$

IV. PROPOSED MODIFICATION ON FRACTAL CODING METHOD

In this section, the entire process of fractal coding based on the proposed modifications are discussed as the Encoding and Decoding Process diagram shown in Figures 3, and 14.

A. BLOCKS PREPARATION

First, we chose the standard Zelda gray-scale image (ID-32 of Set-A), and then we resized it as shown in Figures 8, and 7, where range and domain blocks are Center and diagonally symmetric. Reference [10] mentions the thinning algorithms yields suitable result in, considering 8-neighbors around any random pixel regarded as a central pixel. Apart from, [79] states that researchers mainly uses the thinning algorithms to extract significant features, layer by layer from a digital image. Following this, we then looked at $p_c = P(x+1, y+1)$ as a central pixel of eight neighbors of any block in a pixel coordinate structure for the purpose of fractal image coding. That is, for any pixel, p_c has two upright and two horizontal neighbors as $N_4(p)$ and four diagonal neighbors as $N_D(p)$ as shown in Figure 6. In an image block, the one CP, and the eight neighbors around CP with pixels intensity $[0, 2^b - 1 \mid b = 1, 2, \ldots 8]$ of each represents intensity of any position with in the block. In spatial coordinate, their position vectors are follows:

$$N_4(p) : (x, y+1), (x+2, y+1), (x+1, y), (x+1, y+2)$$
$$N_D(p) : (x, y), (x+2, y+2), (x+2, y), (x, y+2).$$

Subsequently, the 8-neighbors as $N_4(p) \cup N_D(p) = N_8(p)$. A spatial function in the image plane relates pixel intensity and position. Mathematically, the central pixel, p_c, $[0, 2^b - 1] = f(x+1, y+1)$, helps to binarize the block intensity using the intensity of as threshold. We made two copies of resized Zelda standard image, and partitioned both as range and domain blocks of size $\alpha \times \beta$ and $(ps \times \alpha) \times (ps \times \beta)$ respectively, as shown in Figures 7 and 8, where $\alpha = \beta = 2d + 1$ and $ps = 2p + 1$. We applied 3×3 pixels as the size of the range block to get the central pixel of each block, $d = 1, 2, 3, 4$, and $p = 1, 2, 3, 4$ depending on image size. Algorithm 1 supports the code to prepare initial blocks. Besides that we require five more steps to process the blocks according to Algorithms 2, 3, 4, 5, and 6

B. POOLING METHOD

Pooling, the image spatial contraction method (ISC) are mainly two types; Average Pooling and Max Pooling, where the first one, commonly used in the Base Line Method proposed by Jacquin.

Algorithm 1 Domain and Range Blocks Pre-Processing

```
1:  procedure Split($I$)                                   ▷ $I = V \times H$ pixels
2:      Read: $I$                                          ▷ shown in Figure 10(OI-1)
3:      if $V = H$ then
4:  loop:
5:          Split $I$                                      ▷ $size\ of\ domain\ blocks = ps \times (2d+1) \times ps \times (2d+1)$, $ps = 2p+1, p = 1, 2, 3, 4$
6:          goto loop.
7:          close;
8:      Write: The domain blocks                           ▷ D, shown in Figure 10(D)
9:      repeat
10: loop at 4: for Range blocks ▷ $block = (2d+1) \times (2d+1)$, where $d = 1, 2, 3, 4$, in Figure 10(R)
11:     until the total range blocks $= \frac{H}{(2d+1)} \times \frac{V}{(2d+1)}$
12:     close;
13:     Write: Range blocks                                ▷ Range blocks, R,,shown in Figure 10(R)
14:     return
```

Algorithm 2 Spatial Contraction of Domain Block by Using Max-Pooling

```
1:  procedure Max-Pooling(D, ps)                           ▷ Geometric Contraction
2:      Read: The domain blocks ▷ D, from algorithm 1 and shown in Figure 10
3:      $tdb^2 = [(vs/srb)/ps]^2$;                         ▷ Total of domain blocks $(tdb)^2$
4:      for $i \leftarrow 1$ to $srb(d) \times tdb$ do     ▷ $srb(d)$, the size of range block
5:          for $y \leftarrow 1$ to $srb(d) \times tdb$ do
6:              Max-Pooling                                ▷ spatial contraction using (13), in total $= \frac{V}{(2d+1)} \times \frac{H}{(2d+1)}$ as shown in Figures 9, and 11
7:      Write: Contracted domain blocks,                   ▷ AMP-1, shown in Figure10, and 11 right part
8:      return
```

Algorithm 3 Eight-Isometrics Mapping on Domain Blocks

```
1:  procedure Isometric-Mapping(D, $\omega_{ro}$, $\omega_{re}$)   ▷ As discussed in IV-C
2:      Read: Contracted domain blocks, D                  ▷ shown in Figure 10(AMP-1) from algorithm 2
3:      Apply the isometric mapping                        ▷ for each block geometrically using ( 18, and 19 on block matrix
4:      Write: Eight transformed domain blocks, by using $T_1 \dots T_8$ shown in Figures 10, and 12.
5:      return
```

Algorithm 4 Apply Contrast Mapping on Domain Block

```
1:  procedure Contrast-Mapping($T_1(D) \dots T_8(D)$)     ▷ Numeric contraction applied on Pixels result in contrast
2:      Read: Eight transformed domain blocks, $T_1 \dots T_8$   ▷ shown in Figures 10, and 12 from algorithm 3
3:      Calculate $\hat{s}_q$                              ▷ the value of $\hat{s}_q$ for each block as shown in (24)
4:      create $S$                                         ▷ a set $S = \{av(\hat{s}_q) - std(\hat{s}_q), av(\hat{s}_q), av(\hat{s}_q) + std(\hat{s}_q)\}$
5:      for $\tau \leftarrow 1$ to 8 do
6:          for $n \leftarrow 1$ to $S(n)$ do
7:              Construct Domain Blocks, $D_c = D_\tau \times S(n)$   ▷ Each block results in $D_\tau \times S(n)$=8 $\times$ 3, $c = 1\ to\ 24$, and $\tau = 8$
8:      Write: Total domain blocks $= \frac{M}{ps \times (2d+1)} \times \frac{M}{ps \times (2d+1)} \times \tau \times S(n)$   ▷ One of eight domains, $(T_\tau(D))$ only, and $(T_\tau(D))$ reproduces $T_\tau(D) - 01$, $T_\tau(D) - 02$, $T_\tau(D) - 03$ shown in Figures 10, where $\tau = 1$. Thus total of three, $\hat{s}_q \in S$ and eight, $\tau$
9:      return
```

Algorithm 5 Calculating Offset for Range Block

```
1:  procedure Offset(R)                                    ▷ Offset improves brightness
2:      Read: Each range block                             ▷ Total of $\frac{M}{(2d+1)} \times \frac{M}{(2d+1)}$, from algorithm 1
3:      Calculate $g_q$                                    ▷ the value $g_q$ for each block using (25) as a part of optimization.
4:      Write: Block offset values, $g_q$ for each block, in total $= \frac{M}{(2d+1)} \times \frac{M}{(2d+1)}$
5:      return
```

Algorithm 6 Reprocessing Range Block (RRB)

```
1:  procedure RRB($g_q$, $R$)                              ▷ Reprocessing Range Block(RRB)
2:      Read: Block offset value, $g_q$ from algorithm 5
3:      Construct $R(new)$                                 ▷ Apply the value of $g_q$ to each block to construct new range blocks, $R(new) = R(old) - g_q \times ones(size(R))$
4:      Write: Reprocessed range block, $R(new)$, in total $= \frac{M}{(2d+1)} \times \frac{M}{(2d+1)}$
5:      return
```

TABLE 2. The proposed method used three sizes of images and blocks: Total blocks produced in the proposed and some known methods.

Proposed Method		Known Method		% difference of Range Blocks
ImS-BlS	Range Blocks	ImS-BlS	Range Blocks	
135-3	2025	128-2	4096	50.56
135-5	729	128-4	1024	28.81
270-5	2916	256-4	4096	28.81
270-9	900	256-8	1024	12.11
540-5	11664	512-4	16384	28.81
540-9	3600	512-8	4096	12.11
		Overall blocks difference		26.87

1) ISC BY MAX-POOLING METHOD

Reference [51] experiences that max-pooling has been found to work better than average pooling for computer vision tasks such as image classification. On the other hand, the Max-pooling measures the pixel value in each block as

x, y	x, y+1	x, y+2
x+1, y	x+1, y+1	x+1, y+2
x+2, y	x+2, y+1	x+2, y+2

=

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

FIGURE 6. Eight neighboring pixels of CP.

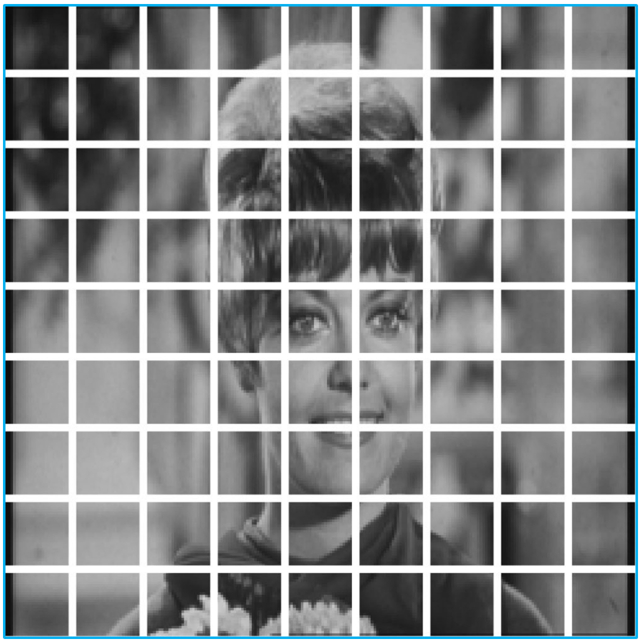


FIGURE 7. Domain block windows.

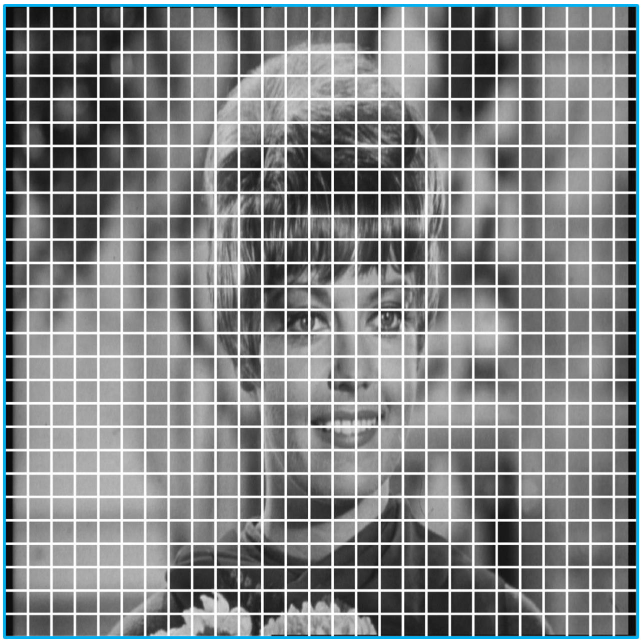


FIGURE 8. Range block windows.

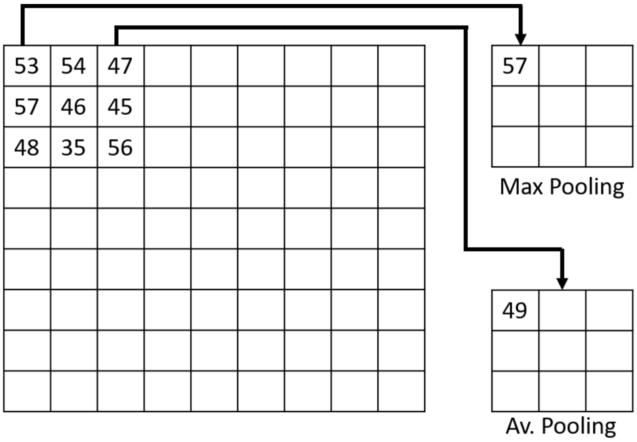


FIGURE 9. Max and average pooling.

the maximum for every function map, Mathematically,

$$I_{max} = max\left(\left[x_{x,y}\right]\right) \tag{13}$$

Reference [11] claims that max-pooling has become popular recently because of its better performance and is suitable for sparse images. All the comprehensive research on Jacquin's fractal transform code, however, used average pooling when we first thought about moving to Max-pooling by encouraging its benefits. We used the Algorithm, 2 to get spatial contraction in domain block.

2) SIZE OF MAX-POOLING

Reference [112] displays both cases of even and odd pooling sizes based on the location information, turning it into displacement function. Zheng knows, in his study, that in even cases all units have displacements in both horizontal and vertical directions. However, in the odd case, Zheng found that the translation would be zero in the horizontal and vertical directions, respectively, if the maximum value would contain in the central row or column. What Zheng has mentioned is nothing more than symmetric pixels and their consistency. Eventually, the odd pooling size returns image data with a peak centered around the central pixel, where even pooling size will not be regarded as consistent when the researcher needs to maintain the peak positions during pooling. So, if researchers think of convolution as an interpolation between the given pixels and a center pixel, by pooling even sizes, they can not interpolate to a center pixel. For an odd-sized pooling in CNN (ConvNet), each of the previous layer pixels will be symmetrically around the output pixel. In the Fractal Image Coding system, the Average Pooling size has been an even in size as 2 since the time the baseline method has started. However, In our study, we used an odd in size as $ps = 2p + 1$, and reduced $[ps \times (2d + 1)] \times [ps \times (2d + 1)]$ block size to $[2d + 1] \times [2d + 1]$ block size by using max-pooling, for illustration, every 3×3 pixels block condenses to 1×1, that means the spatial contraction factor is $\frac{1}{3}$ rather than $\frac{1}{2}$, which is commonly used by all researchers. We need to switch to transform the blocks as follows after the contraction has done. Pooling size (ps) as a contraction factor, one can use in the Algorithm, 2. Besides, some more theoretical ideas about how to build an automated self-regulating system need to demonstrate as follows;

C. DOMAIN BLOCK's TRANSFORMATIONS

The overall transformation expressed in (7) can be shown in the form of automated self-regulating system by using the (14):

$$T\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & \rho & 0 \\ 0 & 0 & s \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e \\ f \\ g \end{pmatrix} \tag{14}$$

where (x,y) is the arbitrary spatial coordinates of pixels, and z is the pixel intensity for corresponding spatial coordinates. The parameters a, b, c, ρ are responsible for geometric transformations, s for contrast adaption, e and f are the translation vectors for luminance adjustment, and the Figures 12 and 10 show the processes, separately for the single domain as well as full image respectively.

1) GEOMETRIC TRANSFORMATION

To transform domain blocks, one can use eight Isometric-Transformations (ITs) based on the alternate form of (14). The Equations (15), and (16) are the vector operators to get transformed blocks and implemented by using Algorithm 3.

$$T_{ro} = \begin{bmatrix} cos\omega & -sin\omega & 0 \\ sin\omega & cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$T_{re} = \begin{bmatrix} cos2\omega & sin2\omega & 0 \\ sin2\omega & -cos2\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Here, for rotation $\omega_{ro} = [0°, 90°, 180°, 270°]$ and for reflection $\omega_{re} = [0°, 45°, 90°, -45°]$, and these eight values of ω transform each domain into eight domains. Considering to the homogeneous coordinates system, we let the intensity vector, $z = 1$ so that the column vector becomes,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{17}$$

According to [7], and [104], the (14) should have at least three properties as discussed in III-A. In this section, we addressed only geometric aspect by the following equations, (18), and (19) by using (15), and (16);

$$T_g(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}) = \begin{bmatrix} cos\omega & -sin\omega & 0 \\ sin\omega & cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{18}$$

and

$$T_g(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}) = \begin{bmatrix} cos2\omega & sin2\omega & 0 \\ sin2\omega & -cos2\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{19}$$

The magnified transformed domain block results in according to Figure 12.

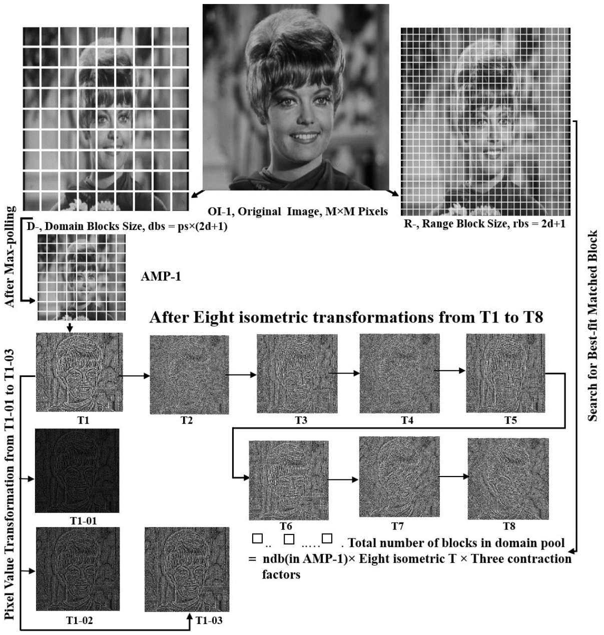

OI-1, Original Image, M×M Pixels
D-, Domain Blocks Size, dbs = ps×(2d+1)
R-, Range Block Size, rbs = 2d+1
After Max-polling
AMP-1
After Eight isometric transformations from T1 to T8
T1
T2
T3
T4
T5
T6
T7
T8
T1-01
T1-02
T1-03
Pixel Value Transformation from T1-01 to T1-03
Search for Best-fit Matched Block
□.. □ .....□ . Total number of blocks in domain pool
= ndb(in AMP-1)× Eight isometric T × Three contraction factors

FIGURE 10. For a whole image blocks, eight isometric transformations with 3 scale factors is taken to adjust contrast, T1-01, T1-02, T1-03.

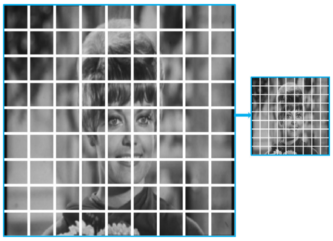


FIGURE 11. Domain blocks max-pooling.

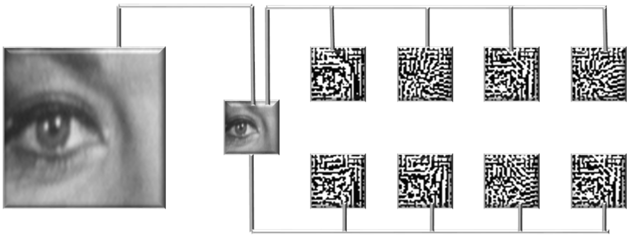


FIGURE 12. For a domain, eight isometric transformations:T1. Invariant transform, T2. 90 degree CCW rotation, T3. 180 degree rotation, T4. 270 degree CCW rotation, T5. Reflection on horizontal axis, T6. Reflection on vertical axis, T7. Reflection on $y = -x$ line, T8. Reflection on $y = x$ line.

2) IMAGE CONTRAST AND BRIGHTNESS ADJUSTMENT

From (14), the contrast, s and brightness, g of a gray-scale image can be calculated by introducing the third dimension, z in the spatial coordinates. The (14) can be rearranged as follows;

$$\Rightarrow T\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \\ sz + g \end{pmatrix}. \tag{20}$$

It is clear here we are having an equation for depicting the intensity function,

$$T(z) = sz + g \tag{21}$$

where, the variable z and $T(z)$ are the domain and range block intensity respectively. We can estimate a linear regression line

$$\hat{T}(z) = \hat{s}_q z + g_q \tag{22}$$

and thus we get the following error,

$$E = \sum\{T(z) - \hat{T}(z)\}^2 = \sum\{T(z) - g_q - \hat{s}_q z\}^2 \tag{23}$$

where $z = f(x, y)$, however, to optimize the error in (23), one can take partial derivatives with respect to g_q and $\hat{s}_q$ and solve the equation to get $\hat{s}_q$, g_q

$$\hat{s}_q = \frac{\lambda \sum z \times \hat{I}(z) - \sum z \sum \hat{I}(z)}{\lambda \sum z^2 - (\sum z)^2} \tag{24}$$

and

$$g_q = \frac{1}{\lambda}\Big[\sum \hat{I}(z) - \hat{s}_q \sum z\Big] \tag{25}$$

where λ is the number of pixels in each block and the minimum error in Equation 23 will support to match between the blocks pair as D_{2d+1}, and R_{2d+1}, where the pixel intensity of range block is as follows:

$$R_{2d+1} = \hat{s}_q * D_{2d+1} + g_q. \tag{26}$$

Algorithms 4 and 5 are expected to bring this theoretical aspect into effect.

D. FINDING CENTRAL PIXEL OF EACH BLOCK

First, we must find the central pixel for each block during the block feed process that was achieved by the function,

$$I_{\frac{\alpha+1}{2} \times \frac{\beta+1}{2}} = B(\frac{\alpha+1}{2}, \frac{\beta+1}{2}). \tag{27}$$

where, $\alpha = \beta = 2d+1$, where $d = [1, 2, 3, 4]$. Algorithms 7 suggests executing this approach.

E. THE BLOCK PIXEL BINARIZING PRINCIPLE

Reference [66] notes that since the optical character analysis, identification, and classification of natural images involves a prior image binarization, the implementation of classical global thresholding methods in such a situation makes it difficult to maintain the visibility of all characters, and claims that regional binarization is thus substantially important. Eventually, they states that the image binarization is one of the most important preprocessing measures contributing to a substantial decrease in the quantity of information sent for further analysis and to an improvement in its speed. Following this, we set a function to convert the intensity values of pixels in each individual block to logical binary digits using the following function.

$$\Psi(I_{x\prime \times y\prime}) = \begin{cases} 1 & \text{if } I_{x\prime \times y\prime} \geq p_c \\ 0 & \text{if } \textit{otherwise} \end{cases} \tag{28}$$

Algorithm 7 CPs for Domain and Range Blocks

```
procedure CPE($D(new)$,$R(new)$)    ▷ Central Pixel Extraction (CPE)
    Read: New domain and Range blocks from algorithms 4, and 6
    if $Block = Domain\ block$ then
        Find $CPs$, $C_p(D) = B(\frac{\alpha+1}{2}, \frac{\beta+1}{2})$    ▷ central pixel of the block, where $\alpha = \beta = 2d + 1$
    else
        Find $CPs$, $C_p(R) = B\{\frac{\alpha+1}{2}, \frac{\beta+1}{2}\}$ ▷ $CPs$, $C_p(R)$ in (28)
    Write: The total CPs, $C_p(D)$, and $C_p(R)$    ▷ in total $\frac{M}{ps\times(2d+1)} \times \frac{M}{ps\times(2d+1)} \times \tau \times S(n)$, and $\frac{M}{(2d+1)} \times \frac{M}{(2d+1)}$ respectively
    return
```

where $I_{x\prime \times y\prime}$ = Local Pixel Intensity of the blocks in $N_8(p)$, and inclusive p_c, the Central Pixel and $x\prime \times y\prime$ is the array of pixels in block. Algorithm 8 recommends this strategy to be executed.

F. DEMONSTRATION OF BLOCK INTENSITY SHIFT USING CENTRAL PIXEL

$\Psi^B_{\alpha\times\beta}$ is a logical binary-valued $\alpha \times \beta$ order matrix, and it is formed by using (27), and (28)

$$\begin{aligned}\Psi^B_{\alpha\times\beta} &= f(I_{\alpha\times\beta}) \\ &= f(\begin{bmatrix} I_{1\times 1} & \cdots & I_{1,\beta} \\ \vdots & I_{\frac{\alpha+1}{2},\frac{\beta+1}{2}} & \vdots \\ I_{\alpha,1} & \cdots & I_{\alpha,\beta} \end{bmatrix}) \\ &= f(\begin{bmatrix} 22 & \cdots & 25 \\ \vdots & 36 & \vdots \\ 27 & \cdots & 36 \end{bmatrix}) \\ &= \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 1 \end{bmatrix}. \end{aligned} \tag{29}$$

G. HUNTING DOMAIN BLOCKS FOR CORRESPONDING RANGE BLOCKS

When the algorithm 9 begins to hunt domain blocks for corresponding range blocks, it changes the pixel intensity of both blocks into binary values and then calculates the optimal error between the domain and range blocks. This is performed by employing the most favorable values of o and s. Thus, we have a collection of the most advantageous domain blocks for every corresponding range block as shown in Figure 13.

H. FRACTAL IMAGE PRODUCTION

Algorithm 10 can produce a fractal-image displayed in Figure 20 from any arbitrary image of the same size as the original shown in Figures 4.

Algorithm 8 Block Pixel binarizing(BPB): The Domain and Range Blocks

```
procedure BPB(($C_p(D)$,$D(new)$),($C_p(R)$, $R(new)$))  ▷ BPB using $C_p(D)$, $C_p(R)$ for Domain and Range blocks, respectively
    Read: New domain and range blocks, and corresponding CPs,  ▷ from algorithms 4, 6, and 7
    if $Block\ set = Domain\ block$ then
        Binarize $Block's\ Pixel$ of Domain,  ▷ shown in (28), and demonstrated with Example in 29
    else
        Binarize $Block's\ Pixel$ of Range,  ▷ The CP(block) will check the 8-neighboring pixel intensity of block and itself
    Write: The binarized domain, and Range blocks $D(new)$, and $R(new)$, in total = $\frac{M}{ps\times(2d+1)} \times \frac{M}{ps\times(2d+1)} \times \tau \times S(n)$, and $\frac{M}{(2d+1)} \times \frac{M}{(2d+1)}$, respectively.
    return
```

Algorithm 9 Collection of IFS-Data: By Searching the Best-Fit Domain Block

```
procedure IFS-Data($D(new)$, $R(new)$)
    Read: The latest range and domain blocks with binary intensity from algorithm 8, respectively.
    calculate, $error$
    $error, \mathcal{L} = norm(D(new) - R(new))$
    if $\mathcal{L} < \frac{\varepsilon}{1-\hat{s}_q}$ then  ▷ $\varepsilon$ is any suitable standard error
        preserve, $IFS - Data$  ▷ the corresponding $IFS - Data$, $x_{ifs}$, $y_{ifs}$, $\hat{s}_q$, $g_q$ and $t_{ifs}$
    else
        go to new search
    Write: A complete set of IFS-Data [$x_{ifs}$, $y_{ifs}$, $\hat{s}_q$, $g_q$, $t_{ifs}$]
    return
```

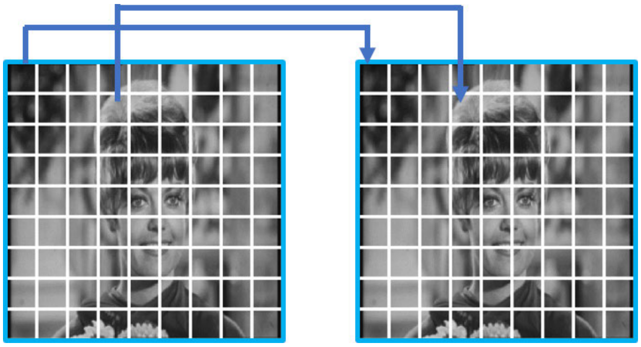


FIGURE 13. Blocks mapping: using Hausdorff Metric.

V. ALGORITHM OF THE METHOD, MPOBPBCPM

We addressed the following algorithms for the proposed method, and we achieved a series of results by running a sequence of codes using Matlab programming through these Algorithms addressed in Section V.

Algorithm 10 Fractal Decoding Algorithm

```
1:  procedure Fractal-Image(IFS-Data)  ▷ Fractal Decoding of IFS-Data to get Attractor Image
2:      Read $AI$,  ▷ Any Arbitrary Image (AI) with the same size $M \times M$ of the original image,  ▷ shown in Figure 11
3:      Load $IFS - Data$  ▷ From algorithm 9
4:      Resize $AI$  ▷ using max-pooling where pooling factor, $\frac{1}{ps}$.
5:      Prepare, $D_p$  ▷ initial preparation of Domain block from dummy image
6:      Transform, $x_{ifs}$, $y_{ifs}$  ▷ Spatial pixel coordinate into $D_p$
7:      Transform, $\hat{s}_q$  ▷ intensity Block-wise intensity transform into each $D_p$
8:      Add, $g_q$  ▷ Vector addition of offset into $D_p$
9:      Apply, $t_{ifs}$  ▷ ISO-metric transforms to build each $D_p$, finally
10:     if input = [$x_{ifs}$, $y_{ifs}$, $\hat{s}_q$, $g_q$, $t_{ifs}$] then
11:         calculate $R_q = \hat{s}_q * D_p + g_q * ones(size(D_p))$.  ▷ IFS-data set Provides all the information
12:     else
13:         Repeat  ▷ Step 6 to 9
14:     Write: $R_q$ as fractal image.
15:     return
```

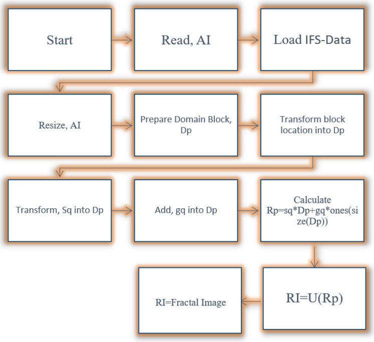


FIGURE 14. Decoding process diagram.

VI. DETAILING OF THE EXPERIMENTAL RESULTS

The writers have demonstrated the encoding speed of the proposed algorithm with the objective quality assessments of the decoded image in this section. In the experiment, we used all sixty-six images from set A and B in Figure 4, and 5.

A. THE EXPERIMENTAL RESULTS ON ENCODING TIME

We have exhibited the encoding time of thirty six different sizes of images with several odd pixels blocks in Figures 15. Table 3 shows the encoding speed, while Figure 16 represents the box-plot of processing time. The improvement of the process time complexity of the proposed method proves the efficiency and overall effectiveness of the scheme through Table 8. The Result of Encoding time of Set B images displayed in Table 4.

TABLE 3. Encoding Time (s) (ET) of several sizes of images and blocks: The final row shows the Mean ET (MET), image s of Set A.

Image ID	135-3	135-5	270-5	270-9	540-5	540-9
1	0.250	0.054	0.172	0.078	0.649	0.265
2	0.172	0.047	0.182	0.141	0.722	0.235
3	0.110	0.043	0.159	0.062	0.688	0.250
4	0.110	0.049	0.169	0.062	0.729	0.311
5	0.110	0.045	0.171	0.078	0.634	0.297
6	0.109	0.045	0.171	0.125	0.647	0.312
7	0.109	0.042	0.168	0.078	0.662	0.297
8	0.110	0.043	0.172	0.063	0.696	0.313
9	0.109	0.040	0.157	0.078	0.683	0.265
10	0.109	0.044	0.157	0.078	0.633	0.250
11	0.188	0.042	0.187	0.063	0.644	0.562
12	0.265	0.044	0.172	0.078	0.698	0.313
13	0.250	0.040	0.156	0.063	0.663	0.266
14	0.265	0.043	0.172	0.094	0.632	0.296
15	0.234	0.040	0.157	0.062	0.665	0.250
16	0.109	0.044	0.172	0.078	0.674	0.531
17	0.110	0.041	0.172	0.094	0.680	0.563
18	0.110	0.043	0.171	0.078	0.829	0.250
19	0.110	0.040	0.172	0.062	0.637	0.250
20	0.109	0.041	0.328	0.078	0.647	0.250
21	0.093	0.041	0.328	0.062	0.747	0.250
22	0.094	0.041	0.219	0.063	0.672	0.295
23	0.110	0.041	0.172	0.079	0.641	0.258
24	0.234	0.041	0.171	0.063	0.648	0.260
25	0.328	0.069	0.157	0.063	0.645	0.233
26	0.278	0.098	0.172	0.062	0.714	0.266
27	0.248	0.089	0.156	0.062	0.669	0.297
28	0.142	0.086	0.156	0.063	0.686	0.266
29	0.094	0.076	0.141	0.046	0.690	0.239
30	0.109	0.087	0.156	0.063	0.642	0.239
31	0.093	0.085	0.156	0.078	0.638	0.245
32	0.109	0.087	0.156	0.063	0.641	0.251
33	0.110	0.079	0.172	0.062	0.703	0.284
34	0.094	0.079	0.141	0.063	0.641	0.233
35	0.109	0.061	0.157	0.063	0.656	0.256
36	0.109	0.043	0.172	0.062	0.669	0.300
MET	0.150	0.055	0.176	0.073	0.673	0.292

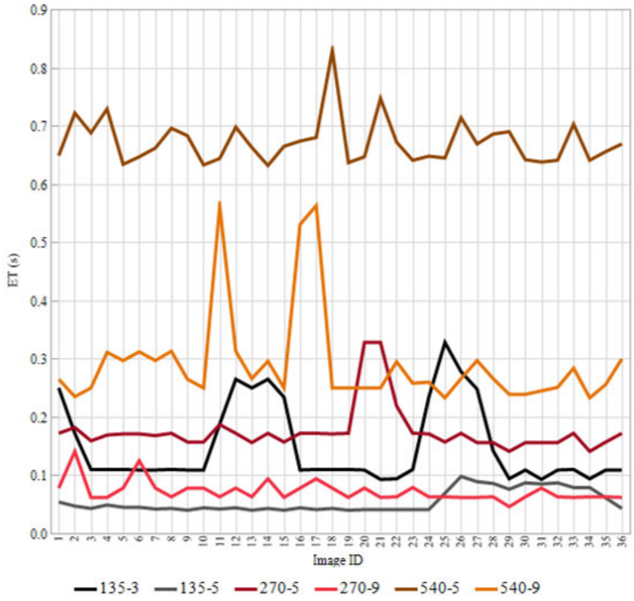


FIGURE 15. Visual display of fractal image encoding time.

B. THE EXPERIMENTAL RESULTS ON OBJECTIVES QUALITY MEASURES

The Algorithm 10 of decoder generated the fractal images of Figure 20 using the corresponding IFS-Data with competitive values of objective image quality shown in Table 5. The competitive results indicate the efficiency of the Algorithms of the proposed method.

TABLE 4. Encoding Time (s) (ET) of another Set B of images and blocks: The final row shows the Mean ET (MET).

Image ID	135-5	270-9	540-9
1	0.042	0.095	0.252
2	0.048	0.093	0.538
3	0.052	0.072	0.239
4	0.048	0.070	0.292
5	0.045	0.080	0.530
6	0.045	0.077	0.492
7	0.042	0.061	0.295
8	0.046	0.076	0.243
9	0.042	0.062	0.246
10	0.042	0.060	0.240
11	0.044	0.062	0.310
12	0.046	0.063	0.248
13	0.046	0.063	0.315
14	0.044	0.074	0.243
15	0.045	0.062	0.312
16	0.043	0.076	0.296
17	0.044	0.117	0.249
18	0.044	0.061	0.245
19	0.041	0.074	0.254
20	0.042	0.063	0.313
21	0.048	0.064	0.260
22	0.045	0.075	0.262
23	0.042	0.067	0.527
24	0.045	0.061	0.259
25	0.042	0.062	0.243
26	0.042	0.061	0.250
27	0.043	0.074	0.254
28	0.049	0.073	0.255
29	0.044	0.062	0.521
30	0.042	0.078	0.314
MET	0.044	0.071	0.310

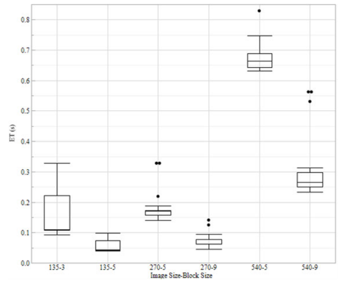


FIGURE 16. Box-plot of encoding time for several sizes of images and blocks.

C. ANALYSIS, COMPARISONS, AND DISCUSSIONS

Thirty-six images were used by authors, and each has three sizes and two block groups shown in Table 2, where we view comparable image sizes, total blocks, and percentage difference in total-blocks. The Figure 15 shows the compatibility of image size with the block size by displaying the nature of fluctuation of encoding time. For instance, $135 - 5$ and

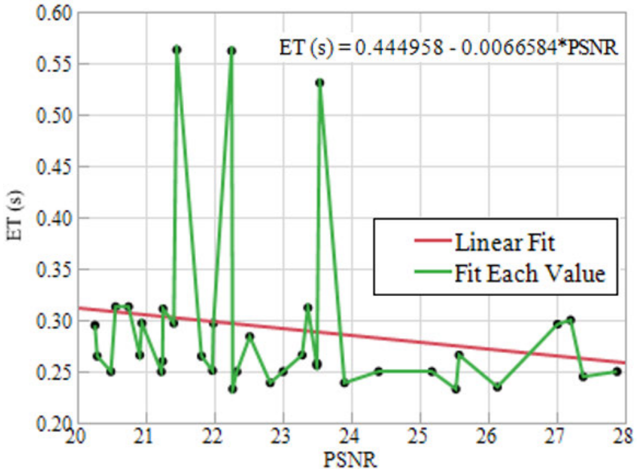


FIGURE 17. ET (s) versus PSNR.

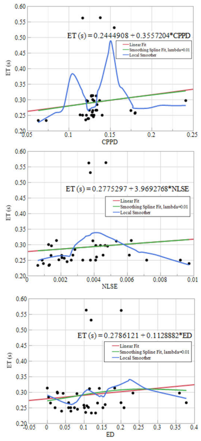


FIGURE 18. ET(s) vs CPPD, NLSE, and ED.

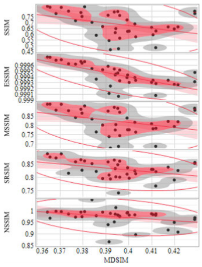


FIGURE 19. Scatter plot of MDSIM versus MSSIM, SRSIM, MSSIM, ESSIM, and SSIM containing linear fit, Non par-density, density ellipse curve.

270 − 9 are more compatible than 540 − 9. Figure 16 shows that 270 − 5 attains minimum interquartile range (IQR) with three outliers while 135−5 has no outlier and 270−9 has two, which are very close to max value. Thus, we can conclude to reprocess pixel optimization further. Table 3 displays the raw data of the encoding time of images of different sizes. Table 3 shows 0.040 (s) and 0.829 (s) respectively minimum and maximum average time for 135 − 5, and 540 − 5, and depicts encoding times (ET) of thirty-six images, where Lena image (Image ID-18) of different sizes are comparable initially with the methods proposed by [4], [34], [44], [89], and [43] and according to Table 8, we find definite improvement according to the size declared in Table 2. Table 6 represents Encoding

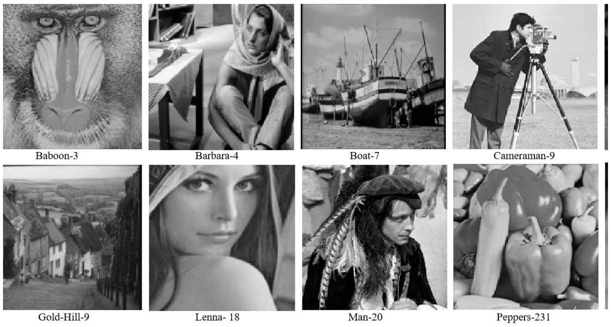

Baboon-3
Barbara-4
Boat-7
Cameraman-9
Gold-Hill-9
Lenna- 18
Man-20
Peppers-231

FIGURE 20. MPOBPBCPM-Fractal decoded image.

TABLE 5. Image quality evaluation tools: Objective quality measures for thirty six Fractal decoded Images, of Set-A using proposed:CPPD-Contrast Per Pixel Deviation, ED-Entropy Difference, ESSIM- Edge Based Similarity Index Metric, FSIM- Features Similarity Index Metric, GMSD-Gradient Magnitude similarity Deviation, IFM-Image Fidelity Metric, MDSIM- Mean Deviation Similarity Index, MSE-Mean Square Error, MSSIM- Multi-scale Structural Similarity Index Metric, NLSE-Normalized Least Square Error, NSSIM- No Reference Structural Similarity Index Metric, PSNR-Peak Signal to Noise Ratio, SFF-Sparse Feature Fidelity, SNR-Signal to Noise Ratio, SRSIM-Spectral Similarity Index Metric, SSIM -Structural Similarity Index Metric, W(PSNR)-Weighted PSNR.

ID	QEMS	135-3	135-5	270-5	270-9	540-5	540-9
1	MSE	52.61887	50.40153	53.41472	51.04983	55.25580	53.78742
2	SNR	14.34185	14.27653	14.34356	14.32423	14.26184	14.31031
3	PSNR	23.80633	21.32212	23.34273	20.91478	25.62258	23.03063
4	WPSNR	22.42786	18.61039	20.32518	17.41810	21.87042	18.95792
5	SSIM	0.73270	0.56301	0.64285	0.51108	0.83802	0.68487
6	MDSIM	0.39097	0.41126	0.39861	0.40763	0.35005	0.39511
7	FSIM	0.79612	0.64921	0.68553	0.58706	0.89054	0.73792
8	ESSIM	0.99957	0.99902	0.99925	0.99872	0.99978	0.99944
9	MSSIM	$-\infty$	$-\infty$	0.90129	0.77741	0.92004	0.82169
10	SRSIM	0.87873	0.76869	0.79106	0.70698	0.93837	0.83021
11	NSSIM	0.99382	0.97831	1.00511	0.99803	0.99211	0.97293
12	CPPD	0.14017	0.16254	0.13705	0.14885	0.12858	0.13246
13	GMSD	0.08310	0.16972	0.15408	0.27972	0.13509	0.25479
14	ED	0.10572	0.24096	0.11168	0.23030	0.06969	0.11515
15	NLSE	0.00305	0.00573	0.00329	0.00620	0.00204	0.00355
16	IFM	0.99889	0.99742	0.99904	0.99757	0.99953	0.99901
17	SFF	0.96288	0.91297	0.92248	0.87157	0.93025	0.89025

TABLE 6. Comparison: PSNR and encoding time with some known methods of different size of Lena image.

MID	Methods	PSNR(dB)	ET(s)
1	[34]	34.560	56.600
2	[26]	30.100	600.000
3	[14]	21.200	82.000
4	[84]	31.030	1.700
5	[90]	34.570	8.000
6	[56]	31.140	3242.000
7	[57]	34.910	3242.000
8	[41]	33.620	6.3
9	[113]	32.740	0.370
10	[54]	32.000	1.77
11	[93]	28.020	39.670
12	[99]	31.230	1.860
13	[58]	32.260	16.330
14	[114]	32.560	4.810
15	[104]	29.760	21.000
16	[96]	35.950	6.640
17	[82]	33.600	4.320
18	Proposed Method	27.881	0.250

TABLE 7. Bits per pixel and compression ratio (round up to whole number) of several known methods.

MID	Methds	bpp	CR
1	[47]	0.6450	12
2	[48]	0.6800	12
3	[6]	0.8000	10
4	[72]	0.5000	16
5	[49]	0.6800	12
6	[68]	1.0000	8
7	[34]	0.2100	38
8	[71]	0.5900	14
9	[64]	0.2900	28
10	[80]	0.5000	16
11	[15]	0.1900	42
12	[26]	0.3200	25
13	[88]	0.2700	30
14	[14]	0.5000	16
15	[28]	2.0000	4
16	[83]	1.3185	6
17	[59]	0.3694	22
18	[18]	0.1088	74
19	[21]	1.5600	5
20	[90]	1.0020	8
21	[92]	0.3844	21
22	[36]	1.2250	7
23	[54]	0.8000	10
24	[99]	0.7886	10
25	[82]	0.7400	11
26	Proposed Method	0.1513	53

TABLE 8. Average % improvement of proposed method in comparing to some known methods.

MID	Methods	Images	BlS	ImS	ET(s)	%Improvement
1	[34]	Lena	4	256	56.600	99.69
2	[53]	Goldhill, F-16, peppers, Lena	8	512	9.103	96.80
3	[14]	Lena, Boat, Baboon	8	512	79.000	99.63
4	[4]	Lena	4	256	150.000	99.88
5	[83]	Lena, Baboon	8	512	49.500	99.41
6	[84]	Lena, Peppers, Baboon, Boats, F16	8	512	2.520	88.43
7	[91]	Lena, baboon, F16, Peppers	8	256	3.800	98.09
8	[89]	Lena	8	256	4.000	98.19
9	[23]	Lena	8	256	1.930	96.24
10	[90]	Lena, Baboon	8	512	8.000	96.35
11	[103]	Lena, Zelda ,		512	56.000	99.48
12	[92]	Peppers,Lena ,F16,Baboon	8	256	6.305	98.85
13	[44]	Lena	8	256	0.510	85.78
14	[36]	Lena, Baboon	-	512	0.368	20.76
15	[29]	Lena,Baboon,Peppers	4	256	837.830	99.98
16	[41]	Lena, Goldhill, Boat,	-	512	6.830	95.73
17	[113]	Lena,Peppers,F16,Baboon	-	256	0.520	86.06
18	[54]	Lena, Baboon	-	512	1.830	84.06
19	[93]	Lena,Peppers, F16, Baboon	8	256	39.670	99.82
20	[114]	Lena,Peppers, F16, Boats	-	256	5.143	98.59
21	[96]	Boats, Lena	4	512	6.075	88.93
22	[43]	Lena	8	512	0.280	10.71
23	[60]	Baboon, F16, Lena, Peppers	8	256	8.004	99.09
24	[82]	Lena, F16, Baboon, Boats	8	512	14.200	95.26
25	[98]	Lena, Peppers, Zelda	8	256	58.770	99.88
26	[106]	Lena, C-man, Peppers, F16, Goldhill	8	256	17.302	99.58
		Average of Percentage Improvement =				89.82

TABLE 9. Several recently known methods on FIC mainly analyzed Lena, Baboon, Bridge, Couple, Crowd, Man, Peppers, Plane Boats, Cameraman images. The proposed method also analyzed all images and even more.

Methods	PSNR	Av. bpp	Av. ET(s)	ImS, Bls
Proposed Method	23.34	0.111	0.18	270,5
	21.00	0.034	0.07	270,9
	25.62	0.111	0.67	540,5
	23.03	0.034	0.29	540,9

Time (s) and PSNR (dB), where PSNR and process time are 27.881 dB and 0.250 (s) for Lena image, however the average PSNR is about 23.03 dB for the same group shown in Table 5. Figure 17 displays ET (s) versus PSNR (dB) plot, and the graph shows a linear fit with a negative correlation, and it proves that better optimization in pixels between Fractal and original images effects time complexity. Figure 18 shows ET (s) versus CPPD, NLSE, and ED plots, where we use linear, spline, and kernel fits. A simple linear fit is a model of a relationship between two continuous variables, while Spline fit is a smoothing spline that varies in smoothness according to the lambda value, as we choose 0.01. The smoothing spline can help to see the expected value of the distribution of dependent across independent. On the other hand, the Kernel Smoother option produces a curve formed by repeatedly finding a locally weighted fit of a simple curve at sampled points in the domain. Implying this method, we can observe the relationship between variables and determine the type of analysis or fit to perform [50]. For both prediction efficiency and model complexity, the mean deviation similarity index (MDSI) metric indicates a competitive balance. At the same time, MDSI(M) is effective, active, and reliable. It performs reliably, for both natural and synthetic images [69]. The Figure 19 represents scatter-plot of No Reference Structural Similarity Index Metric (NSSIM), Spectral Similarity Index Metric (SRSIM), Multi-Scale Similarity Index Metric (MSSIM), Edge Strength Similarity Index Metric (ESSIM), and Structural Similarity Index Metric (SSIM) concerning Mean Deviation Similarity Index Metric (MDSIM). The negative correlation indicates an inverse relation.

For further comparison, Table 9 shows very recent methods for ET (s), PSNR, and bpp. We have compared four models of images to compare all mention methods in Table 9. In comparison to bpp and ET (s) of the proposed method are superior, and PSNR is competitive where the average of IFM, SFF, ESSIM, and NSSIM is 0.99 that indicates image quality is quite good.

Table 8 reflects the encoding time percentage change where the minimum boost is approximately 10.71% compared to [43]. Hu utilizes a 4FFT algorithm for image compression coding scheme based on Fourier transform energy concentration and results in 0.280(s) shown in Table 8. On the other hand, [113] works on Lena image with result of 0.38 (s) and the average encoding time of four images is 0.52(s), while proposed method achieves in 0.078, and 0.073. Reference [113] removed the most inappropriate domain blocks according to each range block to decrease the search space. Before the best matching search, Zhou optimized the mapping error by adjusting the mapping scheme for the sub-blocks based on an image feature while we proposed max pooling, odd pixel block, symmetric central pixel, and binarization of block intensity before the best-matching search and which result in competitive outcomes. Reference [107] uses four types of images, Pirate, Boat, Peppers, and

Living-room with size 512 × 512 and blocks size 8 × 8 results in SSIM 0.89. Reference [55] shows the SSIM, and FSIM values in average, respectively: 0.959, and 0.984. The proposed method achieved the same features resulted in 0.84 and 0.89 average of size 540−5. Besides, the values of MSSIM, NSSIM, SRSIM, and ESSIM are 0.92004, 0.99211, 0.93837, and 0.99211 respectively. These values suggest that the decoded fractal images are outstanding in terms of features, and structures. Image size less than 243 × 243 pixels, MSSIM shows the value of $-\infty$.

VII. CONCLUSIONS AND FURTHER DIRECTIONS OF IMPROVEMENT

The author lifts this scheme to a point where it is substantially comparable to the standard fixed-size block coding method based on full search, after successfully developing the process with multiple changes. By operating on IS0 standard image sets A and B, the author's coding scheme compresses to 0.1513 bits per pixel on an average with a compression ratio of 53 that is still suitable for preserving the image quality shown in Table 7. Unlike the method of Jacquin comprising an exhaustive search strategy with simplified transformation classes, our effortless search procedure requires no domain classes. However, in the improvement and implementation of the research on extensive domain search, the author believes that a great deal of work needs to explore to produce even better outcomes, considering the success of this research. For example, [81] used Lena, Peppers, Cameraman (C-man) of size 256 × 256 pixels with block-size 4 × 4 pixels and results in 0.049(s), while the proposed method achieve it by 0.167 (s). Thus, the author's proposed algorithm will increase its efficiency in combination with other rapid fractal algorithms.

REFERENCES

[1] T. A. Al-Assadi and Z. H. Khalil, "Image compression by using enhanced fractal methods," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 3, no. 1, pp. 231–241, 2011.

[2] N. M. G. Al-Saidi and A. H. Ali, "Towards enhancing of fractal image compression performance via block complexity," in *Proc. Annu. Conf. New Trends Inf. Commun. Technol. Appl. (NTICT)*, Mar. 2017, pp. 246–251.

[3] K. Balaji and K. Lavanya, "Medical image analysis with deep neural networks," in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, Amsterdam, The Netherlands: Elsevier, 2019, pp. 75–97.

[4] B. Bani-Eqbal, "Enhancing the speed of fractal image compression," *Opt. Eng.*, vol. 34, no. 6, p. 1705, Jun. 1995.

[5] M. F. Barnsley, P. Massopust, H. Strickland, and A. D. Sloan, "Fractal modeling of biological structures," *Ann. New York Acad. Sci.*, vol. 504, no. 1, pp. 179–194, Jul. 1987.

[6] J. M. Beaumont, "Advances in block based fractal coding of still pictures," in *Proc. IEE Colloq. Appl. Fractal Techn. Image Process.*, Dec. 1990, pp. 1–3.

[7] K. Belloulata and J. Konrad, "Fractal image compression with region-based functionality," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 351–362, Apr. 2002.

[8] N. Bhattacharya, S. K. Roy, U. Nandi, and S. Banerjee, "Fractal image compression using hierarchical classification of sub-images," in *Proc. VISAPP*, Berlin, Germany, vol. 1, Mar. 2015, pp. 46–53.

[9] S. Bi and Q. Wang, "Fractal image coding based on a fitting surface," *J. Appl. Math.*, vol. 2014, pp. 1–11, Aug. 2014.

[10] L. B. Boudaoud, B. Solaiman, and A. Tari, "A modified ZS thinning algorithm by a hybrid approach," *Vis. Comput.*, vol. 34, no. 5, pp. 689–706, 2018.

[11] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.

[12] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: Multi-way local pooling for image recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2651–2658.

[13] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 111–118.

[14] G. Caso, P. Obrador, and C.-C. J. Kuo, "Fast methods for fractal image encoding," *Proc. SPIE*, vol. 2501, pp. 583–595, Apr. 1995.

[15] H.-T. Chang and C. J. Kuo, "Fractal block coding using a simplified finite-state algorithm," *Proc. SPIE*, vol. 2501, pp. 536–544, Apr. 1995.

[16] H.-W. Chang, H. Yang, Y. Gan, and M.-H. Wang, "Sparse feature fidelity for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4007–4018, Oct. 2013.

[17] S.-J. Chang, S. Li, A. Andreasen, X.-Z. Sha, and X.-Y. Zhai, "A reference-free method for brightness compensation and contrast enhancement of micrographs of serial sections," *PLoS ONE*, vol. 10, no. 5, May 2015, Art. no. e0127855.

[18] Y.-C. Chang, B.-K. Shyu, and J.-S. Wang, "Region-based fractal image compression with quadtree segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, Apr. 1997, pp. 3125–3128.

[19] G.-H. Chen, C.-L. Yang, and L.-M. Po, and S.-L. Xie, "Edge-based structural similarity for image quality assessment," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2, May 2006, p. 2.

[20] K.-L. Chung and C.-H. Hsu, "Novel prediction- and subblock-based algorithm for fractal image compression," *Chaos, Solitons Fractals*, vol. 29, no. 1, pp. 215–222, Jul. 2006.

[21] A. Conci and F. R. Aquino, "Fractal image coding by multi-scale selection based on block complexity," *J. Geometry Graph.*, vol. 3, no. 1, pp. 57–65, 1999.

[22] A. Conci and F. R. Aquino, "Fractal coding based on image local fractal dimension," *Clinics*, vol. 24, no. 1, pp. 83–98, Apr. 2005.

[23] K. M. Curtis, G. Neil, and V. Fotopoulos 2002, "A hybrid fractal/DCT image compression method," in *Proc. 14th Int. Conf. Digit. Signal Process.*, vol. 2, Jul. 2002, pp. 1337–1340.

[24] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud, "Fractal image compression based on delaunay triangulation and vector quantization," *IEEE Trans. Image Process.*, vol. 5, no. 2, pp. 338–346, Feb. 1996.

[25] F. Davoine and J.-M. Chassery, "Adaptive Delaunay triangulation for attractor image coding," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, Oct. 1994, pp. 801–803.

[26] F. Davoine, J. Svensson, and J.-M. Chassery, "A mixed triangular and quadrilateral partition for fractal image coding," in *Proc. Int. Conf. Image Process.*, vol. 3, Oct. 1995, pp. 284–287.

[27] R. Distasi, M. Nappi, and D. Riccio, "A range/domain approximation error-based approach for fractal image compression," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 89–97, Jan. 2006.

[28] F. Dudbridge, "Least-squares block coding by fractal functions," in *Fractal Image Compression: Theory and Application*, Y. Fisher, Ed. New York, NY, USA: Springer-Verlag, 1995, ch. 12, pp. 229–241.

[29] D.-J. Duh, J.-H. Jeng, and S. Y. Chen, "DCT based simple classification scheme for fractal image compression, *Image Vis. Comput.*, vol. 23, no. 13, pp. 1115–1121, 2005.

[30] J. Erfurt, C. R. Helmrich, S. Bosse, H. Schwarz, D. Marpe, and T. Wiegand, "A study of the perceptually weighted peak signal-to-noise ratio (WPSNR) for image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2339–2343.

[31] A. Eskicioglu and P. Fisher, "A survey of quality measures for gray scale image compression," in *Proc. 9th Comput. Aerosp. Conf.*, 1993, p. 4514.

[32] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2959–2965, Dec. 1995.

[33] Y. Fisher, *Fractal Image Compression: Theory and Application*. Berlin, Germany: Springer, 2012.

[34] Y. Fisher, E. W. Jacobs, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression*. Boston, MA, USA: Springer, 1992, pp. 35–61.

[35] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," *Signal Process.*, vol. 29, no. 3, pp. 251–263, Dec. 1992.

[36] S. Furao and O. Hasegawa, "A fast no search fractal image coding method," *Signal Process., Image Commun.*, vol. 19, no. 5, pp. 393–404, 2004.

[37] M. Gharavi-Alkhansari and T. S. Huang, "Fractal-based image and video coding," in *Video Coding*. Boston, MA, USA: Springer, 1996, pp. 265–303.

[38] J. Han, "Speeding up fractal image compression based on local extreme points," in *Proc. 8th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput. (SNPD)*, vol. 3, Jul./Aug. 2007, pp. 732–737.

[39] J. Han, "Fast fractal image compression using fuzzy classification," in *Proc. 5th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 3, Oct. 2008, pp. 272–276.

[40] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[41] C. He, X. Xu, and J. Yang, "Fast fractal image encoding using one-norm of normalised block," *Chaos, Solitons Fractals*, vol. 27, no. 5, pp. 1178–1186, Mar. 2006.

[42] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Computer Vision, Graphics and Image Processing*. Berlin, Germany: Springer, 2006, pp. 58–69.

[43] J. Hu, J. Deng, and J. Wu, "Image compression based on improved FFT algorithm," *J. Netw.*, vol. 6, no. 7, p. 1041, Jul. 2011.

[44] L. Hu, Q.-A. Chen, and D. Zhang, "An image compression method based on fractal theory," in *Proc. 8th Int. Conf. Comput. Supported Cooperat. Work Design*, vol. 1, 2004, pp. 546–550.

[45] J. E. Hutchinson, "Fractals and self similarity," *Indiana Univ. Math. J.*, vol. 30, no. 5, pp. 713–747, 1981.

[46] A. E. Jacquin, "A fractal theory of iterated Markov operators with applications to digital image coding," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 1989.

[47] A. E. Jacquin, "Fractal image coding based on a theory of iterated contractive image transformations," *Proc. SPIE*, vol. 1360, pp. 227–239, Sep. 1990.

[48] A. E. Jacquin, "A novel fractal block-coding technique for digital images," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1990, pp. 2225–2228.

[49] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 18–30, 1992.

[50] JMP. (1976). *Jmp Statistical Discovery From SAS*. [Online]. Available: https:https://www.jmp.com/en_us/events/ondemand/academic/basic-statistics.html

[51] N. Ketkar and E. Santana, *Deep Learning With Python*. Berkeley, CA, USA: Springer, 2017.

[52] N. A. Koli and M. S. Ali, "Lossy color image compression technique using fractal coding with different size of range and domain blocks," in *Proc. Int. Conf. Adv. Comput. Commun.*, Dec. 2006, pp. 236–239.

[53] J. Kominek, "Algorithm for fast fractal image compression," *Proc. SPIE*, vol. 2419, pp. 296–306, Apr. 1995.

[54] T. Kovács, "A fast classification based method for fractal image encoding," *Image Vis. Comput.*, vol. 26, no. 8, pp. 1129–1136, Aug. 2008.

[55] R. S. Kumar and P. Manimegalai, "Near lossless image compression using parallel fractal texture identification," *Biomed. Signal Process. Control*, vol. 58, Apr. 2020, Art. no. 101862.

[56] C.-M. Lai, K.-M. Lam, and W.-C. Siu, "An effieient algorithm for fractal image coding using kick-out and zero contrast conditions," in *Proc. Int. Symp. Circuits Syst.*, vol. 2, May 2003, p. 2.

[57] C.-M. Lai, K.-M. Lam, and W.-C. Siu, "A fast fractal image coding based on kick-out and zero contrast conditions," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1398–1403, Nov. 2003.

[58] G. Li, "Fast fractal image encoding based on the extreme difference feature of normalized block," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, vol. 4, Nov. 2009, pp. 159–163.

[59] J. Li and C.-C. J. Kuo, "Fractal wavelet coding using a rate-distortion constraint," in *Proc. 3rd IEEE Int. Conf. Image Process.*, vol. 2, Sep. 1996, pp. 81–84.

[60] Y.-L. Lin and W.-L. Chen, "Fast search strategies for fractal image compression," *J. Inf. Sci. Eng.*, vol. 28, no. 1, pp. 17–30, 2012.

[61] R. Liu. (2019). *Weighted Peak Signal-to-Noise Ratio*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/3675-wpsnr

[62] S. Liu, Z. Zhang, L. Qi, and M. Ma, "A fractal image encoding method based on statistical loss used in agricultural image compression," *Multimedia Tools Appl.*, vol. 75, no. 23, pp. 15525–15536, Dec. 2016.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[64] G. Lu and T. L. Yew, "Image compression using quadtree partitioned iterated function systems," *Electron. Lett.*, vol. 30, no. 1, pp. 23–24, Jan. 1994.

[65] B. B. Mandelbrot, *Fractals: Form, Chance, and Dimension*, vol. 706. San Francisco, CA, USA: WH Freeman, 1977.

[66] H. Michalak and K. Okarma, "Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumerical character recognition purposes," *Entropy*, vol. 21, no. 6, p. 562, Jun. 2019.

[67] D. M. Monro, "Class of fractal transforms," *Electron. Lett.*, vol. 29, no. 4, pp. 362–363, Feb. 1993.

[68] D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3. Washington, DC, USA: IEEE Computer Society, Mar. 1992, pp. 485–488.

[69] H. Z. Nafchi, A. Shahkolaei, R. Hedjam, and M. Cheriet, "Mean deviation similarity index: Efficient and reliable full-reference image quality evaluator," *IEEE Access*, vol. 4, pp. 5579–5590, 2016.

[70] U. Nandi, "An adaptive fractal-based image coding with hierarchical classification strategy and its modifications," *Innov. Syst. Softw. Eng.*, vol. 15, no. 1, pp. 35–42, Mar. 2019.

[71] M. Novak, *Attractor Coding of Images*. Linköping, Sweden: Linköping Univ., Citeseer, Institute of Technology, 1993.

[72] G. E. Oien, S. Lepsoy, and T. A. Ramstad, "An inner product space approach to image coding by contractive transformations," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1991, pp. 2773–2776.

[73] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[74] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[75] M. Panigrahy, I. Chakrabarti, and A. S. Dhar, "VLSI design of fast fractal image encoder," in *Proc. 18th Int. Symp. VLSI Design Test*, Jul. 2014, pp. 1–2.

[76] T. S. V. Priya, G. V. Sanchez, and N. R. RaaJan, "Facial recognition system using local binary patterns (LBP)," *Int. J. Pure Appl. Math.*, vol. 119, no. 15, pp. 1895–1899, 2018.

[77] M. A. Rahim, M. S. Azam, N. Hossain, and M. R. Islam, "Face recognition using local binary patterns (LBP)," *Global J. Comput. Sci. Technol.*, vol. 13, no. 4, 2013.

[78] N. A. Z. Rahman, R. Ismail, and A. H. M. Shapri, and M. N. Isa, "Peer adjacent mapping with optimum parameters for fractal image compression on medical images," in *Proc. AIP Conf.*, vol. 2203, 2020, Art. no. 020010.

[79] P. S. Rao and K. Yedukondalu, "Hardware implementation of digital image skeletonization algorithm using FPGA for computer vision applications," *J. Vis. Commun. Image Represent.*, vol. 59, pp. 140–149, Feb. 2019.

[80] E. Reusens, "Partitioning complexity issue for iterated function systems based image coding," in *Proc. 7th Eur. Signal Process. Conf. (EUSIPCO)*, vol. 94, 1994, pp. 1–4.

[81] A.-M.-H. Y. Saad and M. Z. Abdullah, "Real-time implementation of fractal image compression in low cost FPGA," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2016, pp. 13–18.

[82] M. Salarian, E. Nadernejad, and H. M. Naimi, "A new modified fast fractal image compression algorithm," *Imag. Sci. J.*, vol. 61, no. 2, pp. 219–231, Feb. 2013.

[83] D. Saupe, "Fractal image compression via nearest neighbor search," Inst. Informatik, Univ. Freiburg Am Flughafen 17, Freiburg, Germany, Tech. Rep., 1996.

[84] D. Saupe, "Lean domain pools for fractal image compression," *Proc. SPIE*, vol. 2669, pp. 150–157, Jan. 1996.

[85] D. Saupe, R. Hamzaoui, and H. Hartenstein, "Fractal image compression: An introductory overview," Inst. Informatik, Univ. Freiburg Am Flughafen 17, Freiburg, Germany, Tech. Rep., 1997.

[86] I.-A. Szoke, D. Lungeanu, and S. Holban, "Image compression techniques using local binary pattern," in *Proc. IEEE 13th Int. Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2015, pp. 139–143.

[87] N. T. Thao, "A hybrid fractal-DCT coding scheme for image compression," in *Proc. 3rd IEEE Int. Conf. Image Process.*, vol. 1, Sep. 1996, pp. 169–172.

[88] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Trans. Image Process.*, vol. 4, no. 6, pp. 832–838, Jun. 1995.

[89] C. S. Tong and M. Pi, "Fast fractal image encoding based on adaptive search," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1269–1277, Sep. 2001.

[90] C. S. Tong and M. Wong, "Adaptive approximate nearest neighbor search for fractal image compression," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 605–615, Jun. 2002.

[91] T.-K. Truong, J.-H. Jeng, I. S. Reed, P. C. Lee, and A. Q. Li, "A fast encoding algorithm for fractal image compression using the DCT inner product," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 529–535, Apr. 2000.

[92] T. K. Truong, C. M. Kung, J. H. Jeng, and M. L. Hsieh, "Fast fractal image compression using spatial correlation," *Chaos, Solitons Fractals*, vol. 22, no. 5, pp. 1071–1076, Dec. 2004.

[93] C.-C. Tseng, J.-G. Hsieh, and J.-H. Jeng, "Fractal image compression using visual-based particle swarm optimization," *Image Vis. Comput.*, vol. 26, no. 8, pp. 1154–1162, Aug. 2008.

[94] M. Verma and B. Raman, "Center symmetric local binary co-occurrence pattern for texture, face and bio-medical image retrieval," *J. Vis. Commun. Image Represent.*, vol. 32, pp. 224–236, Oct. 2015.

[95] J. Wang, X. Lan, Y. Liu, and N. Zheng, "Fractal image encoding with flexible classification sets," *Chin. Sci. Bull.*, vol. 59, no. 14, pp. 1597–1606, May 2014.

[96] Q. Wang, D. Liang, and S. Bi, "Fast fractal image encoding based on correlation information feature," in *Proc. 3rd Int. Congr. Image Signal Process.*, vol. 2, Oct. 2010, pp. 540–543.

[97] W. Wang and F. Chang, "A multi-focus image fusion method based on Laplacian pyramid," *J. Comput.*, vol. 6, no. 12, pp. 2559–2566, Dec. 2011.

[98] X.-Y. Wang and D.-D. Zhang, "Discrete wavelet transform-based simple range classification strategies for fractal image coding," *Nonlinear Dyn.*, vol. 75, no. 3, pp. 439–448, Feb. 2014.

[99] X.-Y. Wang and L.-X. Zou, "Fractal image compression based on matching error threshold," *Fractals*, vol. 17, no. 1, pp. 109–115, Mar. 2009.

[100] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Nov. 2003, pp. 1398–1402.

[101] B. Wohlberg and G. D. Jager, "Fast image domain fractal compression by DCT domain block matching," *Electron. Lett.*, vol. 31, no. 11, pp. 869–870, 1995.

[102] X. Wu, D. J. Jackson, and H.-C. Chen, "A fast fractal image encoding method based on intelligent search of standard deviation," *Comput. Electr. Eng.*, vol. 31, no. 6, pp. 402–421, Sep. 2005.

[103] Y.-G. Wu, M.-Z. Huang, and Y.-L. Wen, "Fractal image compression with variance and mean," in *Proc. Int. Conf. Multimedia Expo.*, vol. 1, Jul. 2003, pp. I–353.

[104] Z. Wu and B. Yan, "An effective fractal image compression algorithm," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling*, vol. 7, Oct. 2010, pp. V7–139.

[105] C. Xing, Y. Ren, and X. Li 2008, "A hierarchical classification matching scheme for fractal image compression," in *Proc. Congr. Image Signal Process.*, vol. 1, May 2008, pp. 283–286.

[106] W. Xing-Yuan, W. Na, and Z. Dou-Dou, "Fractal image coding algorithm using particle swarm optimisation and hybrid quadtree partition scheme," *IET Image Process.*, vol. 9, no. 2, pp. 153–161, Feb. 2015.

[107] C. Xu, Y. Ye, Z. Hu, Y. Zou, L. Shen, X. Liu, and J. Lu, "A primal-dual algorithm for robust fractal image coding," *Fractals*, vol. 27, no. 7, Nov. 2019, Art. no. 1950119.

[108] H. Zhang, B. Yuan, B. Dong, and Z. Jiang, "No-reference blurred image quality assessment by structural similarity index," *Appl. Sci.*, vol. 8, no. 10, p. 2003, Oct. 2018.

[109] L. Zhang and H. Li, "SR-SIM: A fast and high performance IQA index based on spectral residual," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 1473–1476.

[110] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[111] X. Zhang, X. Feng, W. Wang, and W. Xue, "Edge strength similarity for image quality assessment," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 319–322, Apr. 2013.

[112] Y. Zheng, B. K. Iwana, and S. Uchida, "Mining the displacement of max-pooling for text recognition," *Pattern Recognit.*, vol. 93, pp. 558–569, Sep. 2019.

[113] Y.-M. Zhou, C. Zhang, and Z.-K. Zhang, "Fast hybrid fractal image compression using an image feature and neural network," *Chaos, Solitons Fractals*, vol. 37, no. 2, pp. 623–631, Jul. 2008.

[114] Y.-M. Zhou, C. Zhang, and Z.-K. Zhang, "An efficient fractal image coding algorithm using unified feature and DCT," *Chaos, Solitons Fractals*, vol. 39, no. 4, pp. 1823–1830, Feb. 2009.

[115] K. Zou, Q. Wang, and Z. Zhai, "A novel method to improve the quality of decoded images in fractal image coding," in *Proc. 8th Int. Congr. Image Signal Process. (CISP)*, Oct. 2015, pp. 184–188.

MD AHADULLAH (Member, IEEE) received the B.Sc. degree in engineering science, mechanical from the Chittagong University of Engineering and Technology, Bangladesh, and the M.Sc. degree in applied mathematics from UU, Bangladesh. He is currently pursuing the Ph.D. degree in mathematical cryptography and bio-metrics from the Institute for Mathematical Research, University Putra Malaysia, Serdang, Malaysia. His research interests are in applied image processing, cryptography, and biometrics.

SITI HASANA SAPAR received the B.Sc. degree (Hons.) in mathematics, and the M.Sc. and Ph.D. degrees in number theory from Universiti Putra Malaysia. She is currently working as an Associate Professor with the Department of Mathematics, Faculty of Science, Universiti Putra Malaysia. Her research interests are in the areas related to exponential sums and Diophantine equations. She has also reviewed research articles for national and international journals.

NADIA M. G. AL-SAIDI (Member, IEEE) received the B.Sc. and M.Sc. degrees in applied mathematics from the Department of Applied Sciences, University of Technology, Baghdad, Iraq, in 1989 and 1995, respectively, and the Ph.D. degree in mathematics and computer application sciences from Nahrain University, Baghdad, Iraq, in 2003. She joined the Institute for Mathematical Research (INSPEM), University Putra Malaysia (UPM), as a Post Doctorate Researcher from 2008 to 2010, with the research project "Fractals in Cryptography." In 1989, she joined the Department of Applied Sciences, University of Technology, as an Academic Staff Member. She is currently a Professor with the Department of Applied Sciences, University of Technology-Baghdad, Iraq. She is the author of numerous technical articles since 1994. Her research interests include cryptography, fractal geometry, chaos theory, and graph theory.

MOHAMAD RUSHDAN MD SAID received the B.S. degree in mathematics from Louisiana State University, Baton Rouge, LA, USA, in 1981, the M.Sc. degree in mathematics from the University of Louisiana, Lafayette, LA, and the Ph.D. degree in mathematics/cryptography from Macquarie University, Sydney, Australia, in 1998. He is currently an Associate Professor with the Universiti Putra Malaysia, Malaysia.