Available online at sjuoz.uoz.edu.krd

Science Journal of University of Zakho
Vol. 11, No. 1, pp. 119– 126, January-March 2023

journals.uoz.edu.krd

p-ISSN: 2663-628X
e-ISSN: 2663-6298

FAST FULL-SEARCH ALGORITHM OF FRACTAL IMAGE COMPRESSION FOR ACCELERATION IMAGE PROCESSING

Baydaa Sh. Z. Abood [a,*], Hanan A. R. Akkar [a], Amean Sh. Al-Safi [b]
[a] Department of Electrical Engineering, University of Technology, Baghdad, Iraq (eee.20.32@grad.uotechnology.edu.iq)
[b] Department of Electrical Engineering, University of Technology, Baghdad, Iraq (hanan.a.akkar@uotechnology.edu.iq)
[c] Department of Electrical and Electronics Engineering, University of Thi-Qar, Thi-Qar, Iraq (amean.alsafi@utq.edu.iq)

Received: *25 Jan., 2023 /* ***Accepted:*** *15 Feb., 2023 /* ***Published:*** *20 Feb., 2023* ***https://doi.org/10.25271/sjuoz.2022.11.1.1122***

ABSTRACT:
A new processing algorithm based on fractal image compression is proposed for image processing efficiency. An image will partition into non-overlapping blocks called range blocks and overlapping blocks called domain blocks, with the domain blocks generally bigger than the range blocks, to achieve a rapid encoding time. This research introduced a new fast full-search algorithm approach that starts the search for the best matching domain in the range block from the closest points in the range blocks and expands the search until an acceptable match is found or the search is completed to save even more encoding time. The proposed fast full-search approach, despite its simplicity, is more efficient than the standard search method. The search reduction, peak signal to noise ratio, compression ratio, and encoding time of the suggested approach are all examined. The proposed method can encode a 512x512 grayscale Lena image in 0.36 seconds, with a total search reduction of 87% according to experimental results.

KEYWORDS: Image Processing, FIC, Iteration Function System (IFS), Acceleration of Image, deep data learning, Signal Processing.

I. INTRODUCTION

Many corporations now demand that closed circuit television (CCTV) recordings be continually recorded and saved for at least two weeks. Because they offer better flexibility, higher performance, and are easier to deploy. For CCTV applications, wireless-based high resolution cameras are becoming more common. Three major drawbacks of this flexibility are the high storage requirements, bandwidth limits, and power import restrictions **[49].** As a result, image reduction is a critical component of any CCTV system. Fractal image compression (FIC) is an alternate approach in this area due to its great compression efficiency, higher performance, and simplicity. During transmission or storage, using self-similarity between image blocks, an image of 2-D in size is turned into a statistically uncorrelated dataset or fractals **[1]**.

During the encoding phase, an image will partition into non-overlapping blocks called range blocks and overlapping blocks called domain blocks **[2].** The search method looks for the best-matched domain block for each range block. To do so, all domain blocks must be evaluated, and the best-matched domain block with the fewest matching errors is identified.

FIC based on an iteration function system (IFS)**,** ***Barnsley and Hurd*** **[2]** suggested this method first, and then ***Jacquin*** **[3]** expanded it further to create a partitioning iteration function system (PIFS). FIC has a high CR when compared to other lossy techniques, especially when applied to images captured by satellite imagery or aerial photography **[3].** FIC has a variety of applications in other categories of image processing, like character recognition **[4]** and watermarking **[5]**. A high PSNR and a simplistic decoding technique are two further appealing features of FIC **[6], [7].**

FIC typically necessitates a huge number of searches. As a result, Instead of hours, the encoding time is measured in minutes. Reducing the size of the domain block is one of the most straightforward ways to speed up coding. For each range to what it mapped, a spatial limitation on the domain block is used **[8].** Several academics have developed strategies based on various split decision functions to minimize the size of a domain block in the previous decade **[10,11].** Many scholars have employed the variance feature as a decision function for domain block reduction **[9, 12].** The entropy function has recently can used to minimize the domain block as a split decision function **[14, 13].**

The proposed FIC methods can be classified into three types depending on the search technique: (1) full of search **[15]**, (2) partial of search **[16], [17],** and (3) no search **[18].** The full-search methodology is the most time-consuming of these search strategies. That is because the whole image must be searched to get the best-matched domain block for each range block. In general, the smaller the search space, the faster the encoding. On the other indicator, Partial search is the second most time-consuming method for reducing the search space and speeding up the encoding process. To increase performance and reduce the processing time, many authors have introduced approaches for improving the FIC **[19]–[27].** ***Zheng et al.*** **[19]** and ***Wu et al.*** **[23]** employed genetic strategy, whereas ***Wang et al.*** **[24]** used hybrid encoding method using STD and DCT. Further research **[25]** investigated the Hadamard transformation for reducing runtime. In **[22]** and **[26]**, the method of prediction is combined with another fractal methodology for reducing encoding time. Other studies, like as **[20],[21]**, employed the categorization method to speed up the encoding process by partitioning blocks into classes and searching exclusively for blocks in the same class even with search-less algorithms **[28].** This tendency is slightly deviated by the findings of ***Haque et al.*** **[29]**, ***Erra*** **[30]**, and ***Ismail et al.*** **[31],** these scientists used parallel processing instead of serial computation to accelerate the procedure. As a result, the

* Corresponding author

encoding speed has significantly increased**.** ***Saad, et al.*** **[32]** created a new FIC strategy based on partial similarity search, which achieves a considerable speed increase.

This paper proposes a new fast-full search FIC algorithm for encoding high resolution images with size (512x512) pixel for utilizing the fast-full search approach to detect all pixels in a range block, as well as for future hardware design with acceptable results to serve the detecting system.

The rest of this paper is structured as follows: Section **II** looks at related work on video compression and fractal picture coding. The FIC method's encoding and decoding strategy is showed in Section **III.** Following that, Section **IV** describes the methodology used in this study. The proposed algorithm is then described in Section **V**, followed by the experimental results and study findings in Section **VI**. Section **VII** comes to a conclusion. Finally, Section **VIII** summarizes our future plans and defines the research's direction.

2. RELATED WORKS

Multiple published papers in the literature focused on reducing the fractal technique's encoding time using various methods **[16], [24], [28], [33]-[38].** Even though there are some algorithms for speeding up fractal calculations, practically all of these designs presently employ FIC's fundamental parallelism to enhance. Few designs avoid high-complexity operations, lower operation precision, or combine the two to reduce the computational complexity in coding processes.

For instance, ***L. E. George*****,** ***et al.*** **[39]** introduced a fractal approach according to the vector quantization approach using the block indexing technique. The color image blocks utilized to create the codebook using an affine transform. ***S. Zhu, et al.*****[40]** They explored a method where fractal video sequences are coded in the context of region-based functionality to speed up the encoding process. In another work, ***A. G. Baviskar*****,** ***et al.*** **[41]** presented a strategy for fractal image compression, an effective domain search strategy based on feature extraction is presented, which improved compressed image quality while reducing encoding and decoding time. Likewise, but with more clearly defined classes, the authors ***S. V Veenadevi, et al*** **[42]** introduced a proposal method to achieve fractal image compression, using a threshold value and Quadtree decomposition, the original image will divided into non-overlapping parts. For image encoding and decoding, a threshold value and Huffman coding used, these methods used to compress satellite images.

S. B. Dhok, et al. **[43]** published a new method with full search based on motion estimation technique utilizing FFT that uses a circular cross-correlation approach attemptting to optimize the runtime. The correlation-based similarity metric determines the mean subtracted adjusted coefficient for correlation between the range and the domain block. In comparing it with the standard full-search block matching strategy with similar decoded video quality with the same amount of compression ratio. The recommended algorithm can achieve a faster processing speed.

In the same track, ***C. Rawat, et al.*****[44]** presented a simple method for improving compression quality by using the DCT theorem and fractal image compression techniques. The original image can divided into a total of (8x8) non-overlapping blocks. The DCT utilized for all blocks of the image, and then the DCT coefficients are quantized. The zero values eliminated using a zig-zag approach on the block values, which improves compression quality and extracts nonzero data. In ***U. Nandi, et al.*****[45]** created a new rapid classification strategy with quadtree partitioning technique and DCT for image compression. The compression ratio and peak signal to noise ratio (PSNR) of the fractal image compression approach is maintained by this classification procedure, but the compression time is greatly reduced.

S. B. Dhok et al.**[46]** proposed a new coding approach based on Quadtree decompositions to reduce searching time by Using domain block motion corrected estimation error as little more than a threshold to limit incoming block divisions. In the frequency domain, cross-correlation is performed using FFT, which uses three-level quadtree partitions and quick NCC computation as a matching criterion, as well as a single computing approach for all domain blocks rather than individual block-by-block calculations. The addition of rotation and reflection DFT capabilities to the IFFT considerably reduces the encoding time of null extended range blocks.

On the other hand**,** ***V. Chaurasia, et al.*** **[47]** performed a mechanism method in which domain pool size was optimized by changing the overlapping of domain blocks. The number of the domain blocks formed in domain partitioning is represented in terms of image size and range size. It is directly proportional to image size while indirectly proportional to range size. Later, ***K. Jaferzadeh*****,** ***et al.*** **[48]** proposed an approach to make FIC a faster technique by using local features to reduce search space; a new local binary feature is introduced. The extracted feature is used to calculate distinction among range and domain blocks by using Hamming distance method instead of the least-square method.

Recently, ***Saad, et al.*****[49]** the proposed approach using deep data pipelining to implement FIC, in this approach, two same processes can be applied immediately using two units as a processor. Later time for encoding is reduced by adding the innately more the extent of the relationship amid pixels into vicinity places and best matching domain block search is done in the nearby blocks only. The proposed method based on pipelining strategy encoded a large-size image very fast with reasonable PSNR and CR. While*,* ***A. Banerjee, et al.*** **[50]** introduced another FIC algorithm using quadtree partitioning in which the original image is divided into even part and odd part respectively. One part is divided using the quadtree decomposition of a range of thresholds. Then complete encoding of fractal codes is done using Huffman coding. This proposed suggested scheme reduces time complexity, as well as, gives increased CR and acceptable PSNR. Also, ***A.-M. H. Y. Saad, et al*** **.[51]** they investigated a new method where the fractal technique's search time was reduced as compared to a full-fledged dynamic search approach. The dynamic search starts with a domain block nearest to a range block so it has to be coded and expands until it finds a suitably matching domain block or until the completed search.

The effect of using a spatial dynamic search approach instead of a matching threshold strategy on the search reduction of standard full-search of FIC algorithm studied in ***Saad, et al.*****[52],**When using the matching threshold strategy, the dynamic search approach begins with the closest domain and continues until after an acceptable matching block is found or the full image is covered. Closed sub-image blocks are selected over distant image blocks, according to the proposed approach.

The discussion use of fast wavelet transforms to classify data done in **[53]**. ***Wang*****,** ***et al.*** **[54],** on the other hand, devised a fractal image compression algorithm according to quadtree division, neighbor searching, and strategy of asymptotic. ***Wang, et al.*** **[55]** developed a standard deviation feature for dividing the range blocks into two groups, with the first group range blocks encoded by using the particle swarm optimization technique to find a suitably matched domain block, and the other group range blocks encoded directly by storing their average values without searching.

In general, the main challenges of fractal compression systems are compression ratio (*CR*), peak signal to noise ratio (*PSNR*), and Encoding time. From the previous articles, we see that most of the proposed algorithms were introduced to improve one of these parameters on the other .and most of these algorithms are

used to implement hardware management, not just software applications. Nonetheless, for categorization purposes, the majority of these systems necessitate a large number of logic elements and memory block units**[49].** As a result, when used to encode high-quality images, the circuitry becomes even more complex. As a result, it has been considered to design a simple but effective fractal image compression algorithm that does not necessitate significant additional processes. The solution we presented maintains image quality while balancing the connection between computational complexity, compression ratio, and processing time.

3. FRACTAL IMAGE COMPRESSION

3.1 BACKGROUND

Barnsley** and **Hurd **[2]** were the first to propose the Fractal image compression technology. He created the Iterated Function Systems (IFS) theory and proposed the Collage Theorem, which demonstrates how IFS may be used to approximate images. Using Partitioned Iterated Function Systems (PIFS), ***Jacquin*** **[3]** has developed a fully automatic fractal compression technique that eliminates the requirement for human intervention during the compression process. Various fractal image compression versions and enhancements have been developed as of now. Therefore in this paper, we'll look at PIFS; it divides the original image into smaller and larger parts. We assume that the input image with width **M** and with height **N** may both be represented as powers of 2 when provided the image in the rectangular form of an array with size ***MxN***. The input image can be divided in any way you like, but rectangular blocks are an obvious choice. We used square blocks to simplify the partitioning.

In the coding phase, the input image will be partitioned into non-overlapping range blocks **(rxr)** referred to *as **R*** blocks and overlapping domain blocks **(dxd)** that are larger than rang block and referred to *as **D*** blocks. The size of a domain block has always been four times greater than a range block. As for the matching phase, the domain blocks will resample to match the range block size. This is accomplished by replacing the average gray level of each non-overlapping 2x2 pixel in the domain block.

The below equation is the affine transformation ***W*** search process for finding the best matching domain block for each range block[1]:

$$W = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & S \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e \\ f \\ B \end{pmatrix} \qquad (1)$$

Where ***S*** and ***B*** regulate the contrast and brightness, respectively, ***z*** is the pixel's value at the position ***(x,y).*** The symmetry transformation is also defined by the letters(***a, b, c, d, e, f).*** Only the ***S*** and ***B*** parameters are considered in the algorithm to keep it simple. In the meantime, (***a)*** and ***(d***) are both set to 1 to retain the pixels' placements, while the rest of the parameters well set to zero.

A Mean Square Error ***(MSE)*** is a metric for comparing range and domain blocks. Mathematically [1]:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(Ri - (S \times Di + B)\right)^2 \qquad (2)$$

Where N represents the number of range's block pixels and *(Ri, Di)* is the values of intensity for the range and domain blocks at *the ith* position. To obtain optimal matching between the range and domain blocks, the parameters in ***Eq. 2*** must be approximated accurately. The technique of calculating ideal ***S*** and ***B*** values is comparable to differentiating the ***Eq. 2*** and then equivalent the outputs to zero, according to standard optimization theory.

Hence:

$$\frac{\partial(MSE)}{\partial S} = 0, \qquad \frac{\partial(MSE)}{\partial B} = 0 \qquad (3)$$

The ideal value for *B* in this situation could be calculated in the following way:

$$\frac{\partial(MSE)}{\partial B} = \frac{2}{N} \sum_{i=1}^{N} (-1)(Ri - (S \times Di + B)) = 0$$

$$\sum_{i=1}^{N} -Ri + \sum_{i=1}^{N} S \times Di + \sum_{i=1}^{N} B = 0$$

$$B = \frac{1}{N} \sum_{i=1}^{N} Ri - S \times \frac{1}{N} \sum_{i=1}^{N} Di \qquad (4)$$

In the same way, the ideal value for *S* is:

$$\frac{\partial(MSE)}{\partial S} = \frac{2}{N} \sum_{i=1}^{N} (-Di)\big(Ri - (S \times Di + B)\big) = 0$$

$$\sum_{i=1}^{N} -Ri\, Di + \sum_{i=1}^{N} S \times Di^2 + B \times \sum_{i=1}^{N} Di = 0 \qquad (5)$$

By substituting Eq.4 into Eq.5, we get:

$$S = \frac{N \sum_{i=1}^{N} Ri\, Di - \sum_{i=1}^{N} Ri \sum_{i=1}^{N} Di}{N \sum_{i=1}^{N} Di^2 - \sum_{i=1}^{N} Di \sum_{i=1}^{N} Di} \qquad (6)$$

Finally, The MSE is calculated by putting Eqs. 4 and 6 into Eq. 2 as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} Ri^2 + S\,(S \sum_{i=1}^{N} Di^2 - 2 \sum_{i=1}^{N} Ri\, Di + 2B \sum_{i=1}^{N} Di) + B(N \times B - 2 \sum_{i=1}^{N} Ri\,) \qquad (7)$$

After examining all domain blocks for a particular range block, the ***S*** and ***B*** values corresponding to the domain block with the lowest ***MSE*** are saved. Quantize ***S*** and ***B*** values into specific bit sizes, which are typically 5 to 2 bits for ***S*** and 7 bits for ***B***. When the ***MSE*** goes below a certain threshold, the matching procedures are stopped and that's reducing the search space.

3.2 DECODING PROCEDURE

The decoding procedure is an iterative process that begins with estimation and adjusts it until convergence is achieved. Each range block is reconstructed using the following equation in each iteration[32]:

$$R' = S \times D_{i,j} + B \qquad (8)$$

Where ***R'*** represents the reconstructed range blocks while *D (i, j)* is the best matching domain block in the ***ith*** position, while ***B*** and ***S*** are corresponding domain block's scale and offset values, respectively. In the next iterative cycle, the range of the image evaluated at **Eq. 8** is utilized as the domain of a specified image, and this procedure continues until convergence is obtained. To correctly decompress an image, decoding usually takes no more than 10 iterations. The reconstruction's fidelity is defined in terms of ***PSNR***, which is measured as:

$$PSNR = 10 \log_{10} \frac{255 \times 255}{\frac{1}{M \times N} \Sigma (f - f')^2} \qquad (9)$$

Where *f* is the original image, *f '* is the reconstructed image, and *M* ×*N* is the size of the image [32].

4. METHODOLOGY

A comprehensive search strategy for a collection of transformations in the domain and range blocks that translate an image into itself is offered as an efficient fractal image compression scheme. The proposed methodology is shown in figure (1), where the FIC procedure began by partitioning the input image ***(M X N)*** into square non-overlapping range blocks (***r x r)*** referred to as ***R*** and overlapping domain blocks (***d x d***) referred to as ***D*** and that's partitioning depending on the proximity of a block of pixels in an image to the surrounding area.

The overlapping block partitioning for the construction of the domain pool is done with a step-size ***stp*** which can be from ***1*** to ***d***, where the greater ***stp*** value, the fewer domain blocks are constructed. The following formula can be used to calculate the number of domain blocks ***Nd*** for an (***MxN)*** image **[52]:**

$$Nd = \left(\frac{(M-d)}{stp}+1\right)\times\left(\frac{(N-d)}{stp}+1\right) \qquad (10)$$

As domain blocks are often four times bigger than the size of range blocks where (***d = 2r***), each domain block should be geometrically expanded to the range block's size during the recognition process. The average of every 2x2 pixel is used to do compression. The affine transform is used to improve the level of match between the range blocks and the domain blocks and it can change the constricted domain block's intensity values by multiplying each of them with a certain value of contrast scaling ***S*** and then adding the value of brightness offset ***B*** to the output.

Until the values of ***S*** and ***B*** have been computed are they quantized into a specific bit size, which is commonly 5 to 2 bits for ***S*** and 7 bits for ***B***. For decoding requirements, the fractal codes associated with the identified matched ***D*** must be conserved, and ***D*** blocks must be searched and matched one by one for a certain ***R*** in order to find ***D*** with ***MSE*** less than a pre-defined number. ***S, B,*** and the coordinates for finding matching ***D*** (denoted as ***(Xd, Yd)*** for the ***x***- and ***y***-axes) will be recorded in the fractal codes file. The compressed image file will contain all ***R***. As a result, the following calculation can be used to calculate the size of the compression file (***CF size***) **[52]:**

$$CF\ size = NR(S\ bitsize + B\ bitsize + Xd\ bitsize + Yd\ bitsize) \qquad (11)$$

Where ***NR*** is equal to ***[NR = (MxN) / (rxr)]*** and represents the total number of the range blocks in the encoding image. The remaining coefficients represent the bit sizes of the fractal codes. The compression ratio ***CR*** can be computed using (11), where the original file size of the image ***(OF)***, compressed file size ***(CF)***, and ***[CR=OF/CF]***.

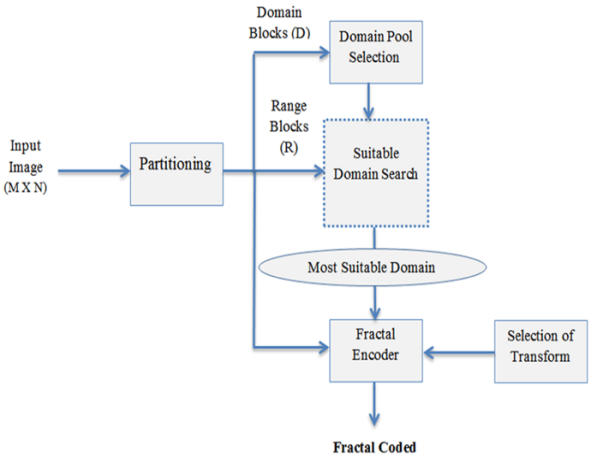


Figure (1): Proposed methodology block diagram

Figure (2) illustrates the fast full-search entire procedure. As shown in the graphic, the search method prioritized the closet domain blocks over the farthest ones. Whereas Figure (2) represents the search condition to range block ***R*** in a center of the image, this is evident that perhaps the nearest domain block to ***R*** is ***D0***, which will be the block that concludes ***R*** inside of it. When ***D0*** does not satisfy ***R***, searching space is pushed to the specified for every direction to include the secondly nearest domain block.

A 3x3 domain block search window will appear as a result. Increase the search window size to 5 then 7 then continue until an adequate matching domain is discovered or the full image is searched. When searching is enlarged, it will begin with domain blocks that locate at the right side of the same horizontal line as ***D0*** and work their way clockwise and repeated, as shown in the graphical construction in the figure below.

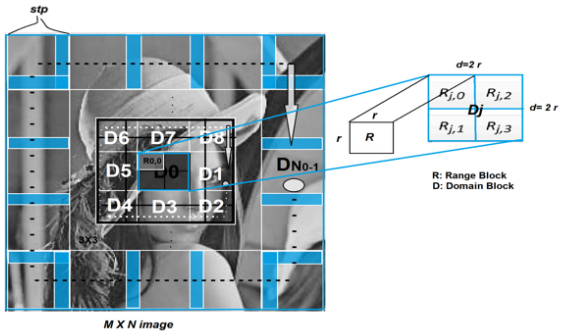


Figure (2): Fast full-search mechanism demonstration.

The flowchart of a fast full-search algorithm for fractal image compression that is utilized to assess our proposed search approaches is shown in Figure (3). The process began with the image initially interred and then the width ***M*** and height ***N*** of the image that well measured.

The size of the range block (***r***), division step-size of the domain blocks (***stp)***, scaling reduction bit-size (***S bitsize***) and the value of main square error (***MSE-thr***) must all be set so that we can use our whole search algorithm technique under various typical coding conditions. To indicate the search processer and encoding phase, these factors are used to calculate the number of blocks for each iteration in range blocks and domain blocks ***NR*** and ***Nd***. After that, for each of the range blocks ***Ri***, Where ***i =*{0, 1,....., NR-1}** and each domain block ***Dj*** where ***j=*{0,1,....., Nd-1}** should be compared one by one, starting with D0. Then the inter-image ***I*** is partitioned into sub-image of (***r x r)*** range block ***(R)*** and ***(d x d)*** domain block ***(D)*** with step-size ***stp***.

The following coefficients ***MSE, B***, and ***S*** are computed at the matching step of the combination of both ***(Ri, Dj)*** *using* equations **(2),** **(4),** and **(6),** respectively. ***Dj*** is the selected best matching domain block for ***Ri*** if ***the MSE*** value is less than the defined ***MSE-thr***. In this situation, the fractal codes for ***Ri*** will be the matching values of ***S, and B,*** as well as the coordinate of the best matching domain *Dj* ***(X, Y)Dj***. Elsewhere, the search under unsatisfied conditions will be continue till the matching domain ***Dj*** with the condition ***MSE < MSE-thr*** is found or searching phase is completed or when all domain block ***Dj*** are detected ***(i.e.; j = Nd -1)*** . Later, if a search has stopped without locating the desired matching domain, the fractal codes of the best effective ***Dj*** are well saved instead. For doing that, the fractal codes of each ***Dj*** discovered at ***MSE < minMSE*** (this is the lowest MSE ever) are well kept in varied temporaries ***(Si, Bi, (Xd, Yd)).*** As a result, when the search is complete, these temporal values will preserve the fractal codes for the best-matched ***Dj,*** which can then be easily stored in a compressed file (***CF***). The method is repeated in the encoded image for each *Ri*.

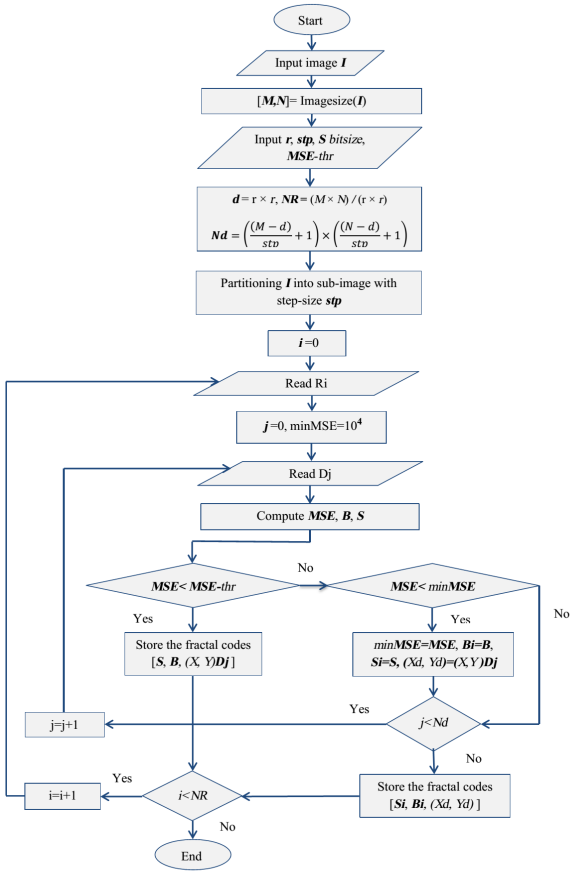


Figure (3): Flowchart of Fast Full-Search Algorithm Approach.

5. PROPOSED ALGORITHM

Algorithm1: Pseudo Code Algorithm for Fast-Full Search Approach

Input: *I, r, stp ,S* bitsize, *MSE*-thr; % Considering *I* square image with *[M,N]* size, where(*M=N*);
Output: Fractal Codes**;**
……Calculate *NR, Nd*;
% number of the range blocks and the domain blocks in a per column or row, respectively.

_Create the range and domain blocks
…….Divide ***I*** into non-overlapping ***r × r*** range blocks {*Ri*};
A **2r×2r** window size with a step of ***stp*** is used to intercepts domain blocks Dj a long ***I***;

_Find the best matching domain based on the similarity measure with proposed fast full search method
For *Ri, where i = {0, 1,. . . , NR − 1}* and *Dj, where j ={0,1,…, Nd −1}*
….Define the lowest *MSE* obtained so far ***(****minMSE);*
……Compute coordinates of *Ri* and *Dj (X , Y),* where *j =0 ;*
……..**For** *C = 1* to *[(N−2r)/stp] and j=1* % *C* decides the search window size
…….....**For** every in *{2C + 1) × {2C + 1)* Obtain coordinates of *Dj* for each window;
% search with window size in a clock-wise directions then beginning from the *Dj* who lies at same horizontal line such *D0* but with starting from the right-side.
……Read *Dj on {xj,yj)Dj ;*

_Do matching process for each pair of *Ri* and *Dj* and end until every range block category has been matched to its domain block category.
…..*Compute MSE, B,S;*
……..**IF** *MSE< MSE-thr* **then** Store fractal code *{S,B,(X,Y)Dj}* of range block *Ri*
Else **IF** *MSE < minMSE* **then** Replace *MSE* with *minMSE* and Store fractal code in temporary variables*{Si,Bi,(Xd,Yd)}*
Else *encode the block.*

6. EXPERIMENTAL RESULTS

The fractal image compression algorithm was implemented in MATLAB R2015b with the standard search method as well as the proposed method to demonstrate the effect of starting the search for a range block from the nearest region over starting from a fixed region regardless of the range block locations. The total number of searches required to encode a grayscale (512X512) Lena image is calculated for both approaches. The results of the suggested and standard methodologies are shown in **Table (1)**. The Peak Signal-to-Noise-Ratio ***(PSNR***) was used to assess compression quality, while the run time or encoding time was utilized to assess search strategy change. As a result, the fewer attempts to choose the optimal domain block, the faster the image can be compressed with a given compression ratio ***(CR).***

As indicated in **Table (1)**, the proposed search strategy required **36406** searches, whereas the standard method required **285677**. This equates to an **87 %** reduction in the total number of searches necessary. Even though the proposed method requires much fewer searches, the quality of the image is increased by **1.5 dB.**

Table (1): *A Comparison between standard and proposed search based on FIC technique for grayscale Lena image.*

	Standard method	Proposed method	Improvement
Total No. Search	285677	36406	87%
PSNR (dB)	29.4	30.92	1.5

Figure (4) shows the decompression results for the Lena image for subjective evaluation. This figure clearly shows that the decoded image produced by the proposed approach is visually preferable to that produced by the standard search approach. There are blocking distortions in the image encoded using the standard approach.

Figure (4): A512x512 gray scale Lena FIC decompression image (a) proposed method (b) standard method

An in-depth examination of the methodology has also experimented on a dataset of grayscale images that can be seen in Figure (5). There are four images in this collection for each **(512x512)** grayscale image (Lena, CameraMan, Baboon, and Peppers). These photos with various textures are utilized to extend the purpose of the study.

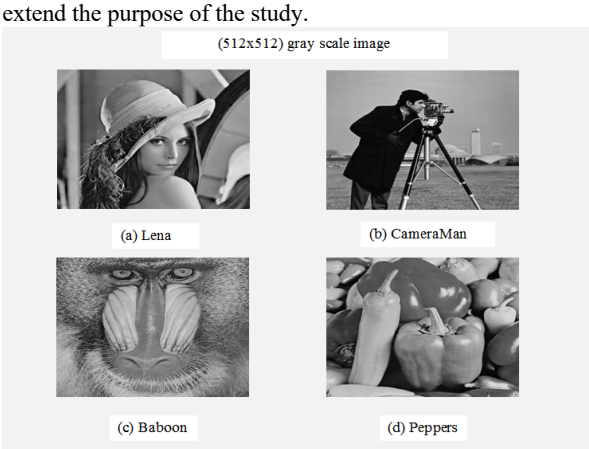
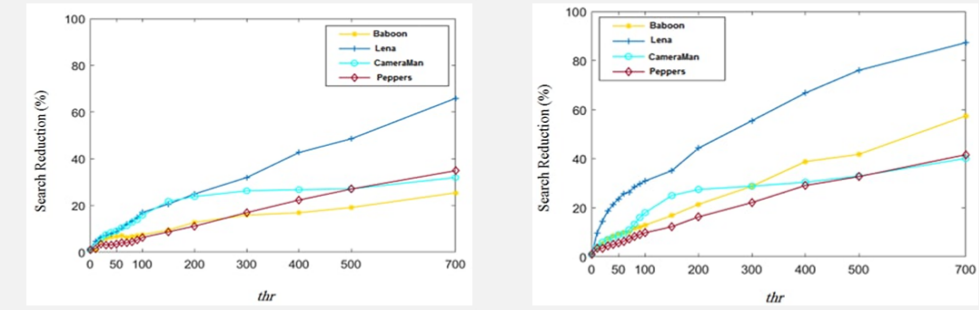


Figure (5): Dataset of four standard images for evaluating the proposal search.

Figure (6) shows the results of testing the fast search strategy with various combinations of ***Sbitsize*** and ***stp***. The degree of similarity between nearby blocks decreases, resulting in a lower preference for close domain blocks over those further away. As a result, the reductions in search numbers derived from a range block (R) of size 8x8 are reduced.

In an examination of the results on the data set in figure (5), the results of combinations shown in Figure (6) reveal a percentage of the search reduction as seen in Figure(6-b). where (***stp*** = 8 and ***S bitsize*** = 5) was the greatest as compared to the results

from the combinations involving (***stp*** = 8 and ***S bitsize*** = 2) in Figure (6-a).The dropping in the efficiency of a proposed searching strategy because of similarity degrees in nearby blocks was decreased typically at range block size and step-size ***stp*** are greater, and when the amount of the contrast scaling ***S*** parameter is lower. So that, the reduction in our search method compared to the standard search method is reduced to the lowest level when we use ***(stp***=8 and ***S bitsize***=2***)*** as demonstrated in figure (6-a).



(a) *stp* = 8, *S bitsize*=2 (b) *stp* = 8, *S bitsize*=5

Figure (6): Fast full search technique curves in respect of search reduction with different *thr*- values, four samples of 512× 512 gray scale images with range block = 8x8.

It was important to compare the performance of the proposed full fast-search scheme to that of other similar studies to complete the evaluation. As a result, Table (2) compares the provided fractal method to five other current fractal compression methods in respect of speed up, ***PSNR***, and the compression ratio (***CR***). The results of the provided approach on Lena image with size (512 x512) are reported in this table, the parameters of fractal codes were ***(thr*** = 700, ***S bitsize***= 5**,** and ***stp = r=8)***. Furthermore, the range block size for the 512x512 grayscale image was (8x8).

Table (2), clearly shows that the presented method takes the least amount of time to encode when compared to the others. Overall, it is obvious that the provided search technique is extremely quick, despite its simplicity.

Table (2): Comparison of the suggested method's performance to that of other fractal approaches.

		Proposed Method	**[16]**	**[53]**	**[48]**	**[54]**	**[55]**
Implementation platform		Intel(R) Core(TM) i5 CPU 1.19GHz	Pentium IV CPU 3GHz	Intel (R) CPU 2.0GHz Core2 Duo	Intel(R) CPU i5 2.5 GHz	Celeron 2.66GHz	Intel (R) CPU 2.0GHz Core2 Duo
Simulation software		MATLAB	C#	MATLAB	MATLAB	C++	C++
Image details		(512x512) Lena image	(512x512) Lena image	(256X256) Lena image	(256X256) Lena image	(256X256) Lena image	(256X256) Lena image
Partitioning method		Fixed	Quadtree	Quadtree	Fixed	Quadtree	Fixed
FIC Parameter	**Encoding time (sec)**	0.36	2.5	56.4	4.2	1.9	6.4
	PSNR	30.92	34.9	25.8	31.5	31.2	25.1
	Compression ratio (CR)	26.18	5.6	15.6	5.3	11.5	27.1

7. CONCLUSIONS

This paper introduced a fast full-search strategy to reduce fractal technique searching time. Rather than starting the search from a specific point in the encoded image without respect for the range block position, the suggested method starts the search from the closest points to the range blocks and continues the search until an appropriate match is found or the search is completed. An experimental finding demonstrated that a proposed searching strategy outperforms the standard search method in all aspects for broad blocks of range for coding conditions and thresholding parameters. When the threshold value is set higher, the gap widens dramatically, and the gap varies depending on the encoding parameters used. The amount of improvement was the biggest reaching over 87% in respect of the reduction of search and 1.5 dB in respect of decoded the quality of an image. In conclusion, it is possible to state that the search mechanism is less with the proposed method and then high speed, fast compress procedure and good image quality with reasonable compression ratio.

8. FUTURE WORK

Using the full-search approach, this fast algorithm search strategy can be employed in the future to provide hardware fast solutions. For example, it can convert the current fast full-searching hardware design to suit the suggested searching scheme. That's a modest adjustment that will result in a significant reduction in encoding time. The hardware implementation of the suggested search technique will not be difficult, since the memory read control unit will simply be changed to read the domain blocks in the sequence described in this research.

ACKNOWLEDGEMENT

The authors wish to acknowledge their gratitude to the staff

members of the Artificial Intelligence laboratory in the Department of the Electrical Engineering/ University of Technology for their help and supporting us during preparing this work.

REFERENCES

[1] Y. Fisher, *Fractal Image Compression: Theory and Application*. New York, NY, USA: Springer-Verlag, 1995.

[2] M. Barnsley and L. Hurd, *Fractal Image Compression*. Natick, MA, USA: A K Peters, 1993.

[3] 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, Jan. 1992.

[4] S. Mozaffari, K. Faez, and M. Ziaratban, ''Character representation and recognition using quad tree-based fractal encoding scheme,'' in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug./Sep. 2005, pp. 819–823.

[5] S. Kiani and M. E. Moghaddam, ''A multi-purpose digital image watermarking using fractal block coding,'' *J. Syst. Softw.*, vol. 84, no. 9, pp. 1550–1562, 2011.

[6] C. E. Martin and S. A. Curtis, ''Fractal image compression,'' *J. Funct. Program.*, vol. 23, no. 6, pp. 629–657, 2013.

[7] M. Polvere and M. Nappi, ''Speed-up in fractal image coding: Comparison of methods,'' *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1002–1009, Jun. 2000.

[8] B. Wohlberg and Gerhard de Jager, "A review of the Fractal Image Compression Literature", IEEE Transactions on Image Processing, vol. 8, No. 12, pp. 1716-1729, Dec. 1999

[9] R. Distasi, M. Polvere, and M.Nappi, " Split Decision functions in Fractal image Coding", Electronics Letters, vol.34, No. 8, pp. 751-753, April 1998.

[10] B. Ramanurthi and A. Gersho, "Classified Vector Quantization of Images", IEEE Transactions on Communication, COM-34, vol. 11, pp. 1105-1115, 1986

[11] E.W. Jacobs, Y. Fisher, and R.D. Boss, "Image Compression: A study of the Iterated Transform Method", Signal Processing, vol. 29, pp. 251-263, 1992.

[12] R. Distasi, M. Nappi, and D. Riccio, "A Range/Domain Approximation Error-Based Approach for Fractal Image Compression", IEEE Transactions on Image Pocessing, vol.15, No. 1, pp. 89-97, Jan. 2006.

[13] B. B. Eqbal, "Enhancing the Speed of Fractal Image Compression", Optical Engineering, vol. 34, No.6, June 1995.

[14] T. Zumbakis, and J. Valantinas, "A New Approach to Improving Fractal Image Compression Times", Proceedings of fourth Intlernational Symposium on Image and Signal Processing Analysis, pp. 468- 473, ISPA 2005.

[15] M. Panigrahy, I. Chakrabarti, and A. S. Dhar, ''Low-delay parallel architecture for fractal image compression,'' *Circuits, Syst., Signal Process.*, vol. 35, no. 3, pp. 897–917, Mar. 2016.

[16] T. Kovács, ''A fast classification based method for fractal image encoding,'' *Image Vis. Comput.*, vol. 26, pp. 1129–1136, Aug. 2008.

[17] H. Wang, ''Fast image fractal compression with graph-based image segmentation algorithm,'' *Int. J. Graph.*, vol. 1, no. 1, pp. 19–28, 2010.

[18] X.-Y. Wang, Y.-X. Wang, and J.-J. Yun, ''An improved no-search fractal image coding method based on a fitting plane,'' *Image Vis. Comput.*, vol. 28, no. 8, pp. 1303–1308, 2010.

[19] Y. Zheng, G. Liu, and X. Niu, ''An improved fractal image compression approach by using iterated function system and genetic algorithm,'' *Comput. Math. Appl.*, vol. 51, pp. 1727–1740, Jun. 2006.

[20] N. Rowshanbin, S. Samavi, and S. Shirani, ''Acceleration of fractal image compression using characteristic vector classification,'' in *Proc. Can. Conf. Elect. Comput. Eng.*, May 2006, pp. 2057–2060.

[21] Y.-G. Wu, M.-Z. Huang, and Y.-L. Wen, ''Fractal image compression with variance and mean,'' in *Proc. Int. Conf. Multimedia Expo (ICME)*, vol. 1, Jul. 2003, pp. I-353-1–I-353-6.

[22] S. Zhu and X. Zong, ''Fractal lossy hyperspectral image coding algorithm based on prediction,'' *IEEE Access*, vol. 5, pp. 21250–21257, 2017.

[23] M.-S. Wu, J.-H. Jeng, and J.-G. Hsieh, ''Schema genetic algorithm for fractal image compression,'' *Eng. Appl. Artif. Intell.*, vol. 20, no. 4, pp. 531–538, 2007.

[24] X. Wang, D. Zhang, and X. Guo, ''Novel hybrid fractal image encoding algorithm using standard deviation and DCT coefficients,'' *Nonlinear Dyn.*, vol. 73, nos. 1–2, pp. 347–355, Jul. 2013.

[25] J. H. Jeng, T. K. Truong, and J. R. Sheu, ''Fast fractal image compression using the Hadamard transform,'' *IEE Proc.-Vis., Image Signal Process.*, vol. 147, no. 6, pp. 571–574, Dec. 2000.

[26] S. Zhu, S. Zhang, and C. Ran, ''An improved inter-frame prediction algorithm for video coding based on fractal and H.264,'' *IEEE Access*, vol. 5, pp. 18715–18724, 2017.

[27] R. da Rosa Righi, V. F. Rodrigues, C. A. Costa, and R. Q. Gomes, ''Exploiting data parallelism on multicore and SMT systems for implementing the fractal image compressing problem,'' *Comput. Inf. Sci.*, vol. 10, no. 1, p. 34, 2016.

[28] X.-Y. Wang, X. Guo, and D.-D. Zhang, ''An effective fractal image compression algorithm based on plane fitting,'' *Chin. Phys. B*, vol. 21, no. 9, Sep. 2012, Art. no. 090507.

[29] M. E. Haque, A. A. Kaisan, M. R. Saniat, and A. Rahman, ''GPU accelerated fractal image compression for medical imaging in parallel computing platform,'' *CORR*, vol. abs/1404.0774, Apr. 2014.

[30] U. Erra, ''Toward real time fractal image compression using graphics hardware,'' in *Proc. Int. Symp. Vis. Comput.*, 2005, pp. 723–728.

[31] B. M. Ismail, B. E. Reddy, and T. B. Reddy, ''Cuckoo inspired fast search algorithm for fractal image encoding,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 4, pp. 462–469, 2018.

[32] A. H. Saad and M. Abdullah, "High-speed implementation of fractal image compression in low cost FPGA," Microprocessors and Microsystems, vol. 47, pp. 429-440, 2016.

[33] W. Xing-Yuan, L. Fan-Ping, and W. Shu-Guo, ''Fractal image compression based on spatial correlation and hybrid genetic algorithm,''*J.Vis.Commun. Image Represent.*, vol. 20, no. 8, pp. 505–510, Nov. 2009.

[34] 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, Nov. 2005.

[35] 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.

[36] X.-Y. Wang, Y.-X. Wang, and J.-J. Yun, ''An improved fast fractal image compression using spatial texture correlation,'' *Chin. Phys. B*, vol. 20, no. 10, Oct. 2011, Art. no. 104202.

[37] K. Jaferzadeh, K. Kiani, and S. Mozaffari, ''Acceleration of fractal image compression using fuzzy clustering and discrete-cosine-transform-based metric,'' *IET Image Process.*, vol. 6, no. 7, pp. 1024–1030, Oct. 2012. [38] 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.

[39] L. E. George and A. M. Kadim, "Color image compression using fast VQ with DCT based block indexing method," in *Springer, 6754*, 2011, pp. 253–263.

[40] S. Zhu, Y. Hou, Z. Wang, and K. Belloulata, "Fractal video sequences coding with region-based functionality," *Elsevier Inc.*, vol. 36, no. 11, pp. 5633–5641, 2012, doi: 10.1016/j.apm.2012.01.025.

[41] A. G. Baviskar and S. S. Pawale, "Efficient Domain Search for Fractal Image Compression Using Feature Extraction Technique," Springer, 2012, pp. 353–365.

[42] S. V Veenadevi and A. G. Ananth, "Fractal image compression using quadtree decomposition and huffman coding," *Signal Image Process.*, vol. 3, no. 2, p. 207, 2012.

[43] R. E. Chaudhari and S. B. Dhok, "Acceleration of fractal video compression using FFT," *IEEE*, pp. 1–4, 2013, doi: 10.1109/ICACT.2013.6710524.

[44] C. Rawat and S. Meher, "A Hybrid Image Compression Scheme Using DCT and Fractal Image Compression.," *Int. Arab J. ,Information Technoly,Zarqa Univ.*, vol. 10, no. 6, pp. 553–562, 2013.

[45] U. Nandi, S. Santra, J. K. Mandal, and S. Nandi, "Fractal image compression with quadtree partitioning and a new fast classification strategy," in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), IEEE.*, 2015, pp. 1–4.

[46] R. E. Chaudhari and S. B. Dhok, "Fast quadtree based normalized cross correlation method for fractal video compression using FFT," *J. Electr. Eng. Technol. Korean Inst. Electr. Eng.*, vol. 11, no. 2, pp. 519–528, 2016, doi: 10.5370/JEET.2016.11.2.519.

[47] V. Chaurasia, R. K. Gumasta, and Y. Kurmi, "Fractal image compression with optimized domain pool size," in *2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC),IEEE*, 2017, pp. 209–212.

[48] K. Jaferzadeh, I. Moon, and S. Gholami, "Enhancing fractal image compression speed using local features for reducing search space," *Springer-Verlag London 2016*, vol. 20, no. 4, pp. 1119–1128.

[49] A.-M. H. Y. Saad and M. Z. Abdullah, "High-Speed Fractal Image Compression Featuring Deep Data Pipelining Strategy," *IEEE Access*, vol. 6, pp. 71389–71403, 2018.

[50] A. Banerjee, U. Biswas, and M. K. Naskar, "Fractal image compression of an atomic image using quadtree decomposition," in *2019 Devices for Integrated Circuit (DevIC), IEEE.*, 2019, pp. 501–504.

[51] A.-M. H. Y. Saad, M. Z. Abdullah, A. M. A. Nayef, and A. S. H. Abdul-Qawy, "An Improved Full-search Fractal Image Compression Method with Dynamic Search Approach," in *2020 IEEE*, pp. 15–18.

[52] A.-M. H. Y. Saad, M. Z. Abdullah, N. A. M. Alduais, and H. H. Y. Sa'ad, "Impact of spatial dynamic search with matching threshold strategy on fractal image compression algorithm performance: study," *IEEE Access*, vol. 8, pp. 52687–52699, 2020.

[53] 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

[54] X.-Y. Wang, F.-P. Li, and Z.-F. Chen, ''An improved fractal image coding method,'' *Fractals*, vol. 17, no. 4, pp. 451–457, Dec. 2009.

[55] 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.