IET Image Processing

Review Article

Computational complexity of fractal image compression algorithm

ISSN 1751-9659
Received on 10th June 2019
Revised 3rd December 2020
Accepted on 16th December 2020
E-First on 26th February 2021
doi: 10.1049/iet-ipr.2019.0489
www.ietdl.org

Richa Gupta[1] ✉, Deepti Mehrotra[2], Rajesh Kumar Tyagi[3]
[1]Computer Science Engineering Department, Amity University, Uttar Pradesh, India
[2]Information and Theory Department, Amity University, Uttar Pradesh, India
[3]Information and Theory Department, Amity University, Haryana, India
✉ *E-mail: richa4483@gmail.com*

Abstract: This study presents insights into the computational complexity of fractal image compression (FIC) algorithms. Unlike JPEG, a fractal encoder necessitates more CPU time in contrast to the decoder. The study examines various factors that impact the encoder and its computational cost. Many researchers have dedicated themselves to the field of fractal encoding to overcome the computational cost of the FIC algorithm. Here, this study offers a look over the approaches in the aspect of time complexity. The automated baseline fractal compression algorithm is studied to demonstrate the understanding of delay in the encoder. The study establishes how various approaches trade-off between the quality of decoder, compression ratio, and CPU time. The experiment section shows the bargain between fidelity criteria of the baseline algorithm.

1 Introduction

Barnsley [1] coined fractal image encoding in 1988. Images encoded using fractal encoder has less storage requirement as compare to other existing alternatives. Aside from the highest compression ratio (CR), a fractal encoding algorithm has tremendous benefits to offer to the industry. It is for this reason, despite numerous years this technology did not wither out. Later an automated version of the fractal image compression (FIC) algorithm was introduced by Jacquin [2]. Later, Fisher [3] presented the idea of a partitioned iterated function system (PIFS) to encode the image, by finding the affine transformation of each image piece, called range block. This breakthrough captured the interest of academia fraternity that reserved their time in drilling each component of basic FIC algorithms to achieve better outcomes. Every fractal image encoding mainly decomposes into three parts: partitioning of an image into two sets of the pool, searching for block pairs and mapping, and final adjustment of the intensity setting of the block to range block.

Each component of the encoder has a reasonable share in time complexity but locating the domain-range pair is the furthermost stimulating. Over the years, various approaches introduced to overcome the computational overbearing of FIC [4–6]. Numerous in-depth surveys and books are written on improving FIC [7–9].

Here, in this paper, we have identified 14 broad categories of FIC enhancement and their computation complexities. These categories of enhancement work on different components of the encoder. Many techniques tend to merge to produce hybrid algorithms [10]. Primarily, various approaches have developed based on classification, quantisation, transformed coding and fixing of the domain block.

The approach [3] worked on the classification of the image block to increase the speed of encoder. It recommends restricting the search to 24 classes of the range block. Each category of image blocks was created using the characteristics of the image block. A block-based encoding of the image further classifies into block transformation and spatial block coding. These two methods used for the reduction of the computational cost of FIC. However, spatial coding cannot entirely exploit the redundancy present in the image. Considering that frequency transform presents the better possibilities in the coding of images. Discrete cosine transform (DCT) and vector quantisation (VQ) is a popular alternative of the spatial block coding of the image. Collectively the block coding concepts showed promising results in accelerating the fractal image coding [11].

The other category of FIC algorithm aimed the second component of the FIC encoder to address the encoding speed of the FIC encoder. The author established the relation between the searching of blocks and the nearest neighbour problem. Nevertheless, the algorithm itself entailed the additional computational cost of $O(N\log N)$ to make a kd-tree construction. Many approaches work on the different viewpoint of an algorithm like partitioning, decoding, mapping to speed up the calculation [12–14]. The algorithm in [3] proposed the HV partitioning scheme as an alternative to the quadtree method. Further, the optimisation of the method in [9] suggested reducing the computational cost of HV partitioning.

Evolution of FIC is an understatement as every year reputed research societies like nature, science direct, IET, IEEE still publishes >1000 articles under the keyword of FIC, encoding, image compression. IEEE alone has published 35,000 papers to date under image compression and the fractal image has a significant contribution to it. The literature encourages the course of this investigation. The present survey is a comprehensive representation of different approaches to establishing the computational cost associated with each one. This study attempts to pinpoint the strength and weakness of various methodologies suggested for enhancing the fractal image encoder. Also, the aim is to provide a platform for further research work in the field of FIC. The objective of the paper is to answer the following research questions.

RQ1: What are the significant factors affecting the performance of the fractal image encoder?
RQ2: What are the existing algorithms to improve the computational cost of FIC encoder?
RQ3: What is the scope of further enhancement of the fractal image encoder?

The rest of the paper is organised as follows. Section 2 lays the foundation of FIC algorithm. Section 3 illuminates the 14 broad enhancement algorithms for fractal image encoding. Sections 4 and 5 examine the fractal parameters, complexities, and fidelity criteria of various algorithms. The experiment is conducted and discussed in the following Sections 6 and 7. Findings and future the scope of the study is concluded in Section 8.

Consider an image f of size M x N. Create a range pool of non-overlapping image blocks v_k of size B x B. Create a domain pool of overlapping image blocks u_k of size 2B X 2B.

Step 1. Range Block Selection: Repeat for every range block $f|v_i$, i = 0, 1, 2,…..N_{Range}-1;
Step 2. Domain pool search: Repeat for every domain block $f|$ u_j , j=0,1,2….., M_{Domain}-1
Step 3. Scaling offset adjustment: Repeat for every a_k, k = 0, 1, 2…. m-1;
Step 4. Luminance offset adjustment: Repeat for every b_l, l= 0, 1, 2…n-1;
Step 5. Distance metric minimisation

$$d(a_k.u_j + b_i, v) = \left\| a_k.u_j + b_l - v \right\| \quad (1)$$

Step 6. Go to step 3.
Step 7. Go to step 4.
Step 8. Store scaling and luminance value for the minimum distance between range and domain block.
Step 9. Stop.

Fig. 1 *Algorithm 1*

Step 1. Choose any image X of random size.
Step 2. Fetch all the affine transformations associated with each range block from the codebook
Step 3. Find the converted image after applying all the stored transformations on image X.
Step 4. Repeat step 3 until images converge to an attractor image.
Step 5. Stop.

Fig. 2 *Decoding algorithm*

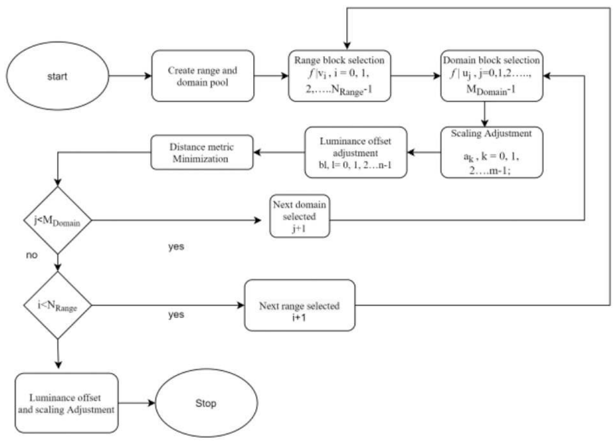

Fig. 3 *Flowchart of baseline fractal image encoder*

2 Preliminaries

2.1 Fractal image encoding

Jacquin [2] introduced a PIFS, which allowed encoding of grayscale images using the FIC algorithm. The fractal encoder partitions the image into two groups called range and domain pool. Following algorithm 1 is a baseline fractal image encoder and decoder for the greyscale image (Figs. 1 and 2).

The baseline FIC algorithm 1 consists of four nested loops. Each step repeated to find a minimum distance metric between two sub-blocks using step 5 in the algorithm. The flowchart showcases the process of image encoding using FIC. The quadruple loop and role of fractal parameters in the encoder are discussed further in the paper (Fig. 3).

The fractal parameters of the algorithm play important responsibility in time consumption of the encoder. The first step and the outermost loop of the algorithm 1 execute until each range block in the pool is mapped. The image is partitioned into N_{Range} number of non-overlapping blocks. The size of the range blocks directly influences the volume of the search space. Smaller blocks are easy to code and need less time to estimate distance metric, but it suffers from computation overhead. Large range blocks exploit redundancy present in the image better as compared to smaller blocks. Step 2 in the encoding process picks each domain block from the pool. Conventionally, the domain block is double the size of the range block. The domain pool is also directly prejudiced by the selection of the size of a domain block. A large number of blocks N_{Domain} require extra time to search the pool to find the appropriate mapping. The number of blocks in the pool depends upon the overlapping allowed. Horizontal and vertical overlying of domain block affect the mapping of domain to range pool. Higher value horizontal overlapping δ_h and vertical overlapping δ_v show the better encoding of the image. However, more overlapping leads to an additional number of land blocks in the pool and is more time-consuming. In steps three and four, the parameters scaling a_k and offset b_l are varied to determine the best match. The range of scaling and luminance offset is based on the number of bits assigned to m and n. High-valued bits result in a better quality of the image. However, it takes more time in encoding the image. *Distance metric calculations*: step 5 in the algorithm is distortion calculation. The time consumption involved is determined by the size of the range block and choice of fractal parameters. Each block is subjected to every combination of the fractal parameter in distortion calculation. Norm ||, || is calculated using the inner product. Sometimes a range block is not covered by any image block from the domain pool then the parent block is further partitioned into smaller parts. There are many possibilities to partition the image block. Each partitioning method influences the time requirement differently.

Here, it is worth noting that each step of the algorithm has led to different approaches that further exercise to improve the computational cost of the FIC algorithm. The following section discusses the strength and weakness of various optimisation approaches.

3 Optimisation approaches of the FIC

The previous section discussed the various factors that affect the performance of the fractal image encoding algorithm. The FIC algorithm is computationally extensive due to large search space and complex distance metric calculation involving an inner product. Here, 14 optimisation approaches are discussed. Every methodology works on a step or more of the baseline algorithm to improve the performance of the encoder. Schemes are discussed in chronological order to showcase the evolution of the optimisation process.

3.1 Block classification

The block classification impacts the search space by grouping it into small manageable parts. A faster mapping is feasible using classified image blocks as compared to random search space. Jacquin [2] proposed a block classification scheme to reduce the encoding time (ET). Fisher's approach [3] proposed a simple method to find isometry of blocks. The algorithm further suggests avoiding exhaustive search (ES) of an isometric set of a block. This approach decreases the ET for one-eighth of the time consumed in full search. However, [3] suffered from the blocking effect on decoding images. The results of the method deteriorate for the lower bit rates. The edge-based process for matching suggested in [11] showed a minor decrease in the superiority of the image. A conventional method mapped range to domain block by minimising the mean squared error (MSE), whereas [11] settled for sub-optimal MSE value. Classifying blocks based on the edge attributes reduces the computation cost significantly [15]. The number of blocks in search space directly affects the time elapsed in encoding. The second factor is the type of partitioning approach used to create the image pool. It is worth observing that smaller blocks are easy to map but are computationally costly to manage.

3.2 Partitioning of image blocks

Quadtree partitioning is relatively simple and has low-cost requirements regarding partitioning representation. However, quadtree partitioning has less flexibility as it allows only change in size corresponding to the local statistic of the image block. Quadtree partitioning suffered from a large number of failed search attempts of mapping in case of large-sized range blocks. Adaptive partitioning regarding arbitrary shape and size suffered from the significant overhead of some bits required to specify the partitioning type for few transformations. Polygon segmentation is slightly less costly regarding the representation of partitioning as only cutting direction and offset is saved. Also, polygon-based segmentation showed better edge rendering as compared to memoryless block-wise partitioning, which suffered from the staircase effect around edges. HV partitioning scheme has an advantage over the quadtree method as it allows the range block adaptively based on its vertical and horizontal edges. The major weakness of HV partitions lies with an increase in some comparisons to map domain blocks in the case range block is significant. Local max–min angle characteristics of Delaunay triangulation helped in reducing localisation of error of pixels present in the triangle. A mixed triangular and quadrilateral partition showed a better CR as compared to FIC based on triangular partitioning alone. Evolutionary FIC performed faster as compared to an error-oriented method since the mapping of range domain is done at the end of evolution. The partitioning of image blocks leads to the mapping of the image block. In the FIC mapping of each range block to appropriate image blocks hugely degrades the encoder's performance. The following approach takes on the alternative measure to map the image blocks.

3.3 Nearest-neighbour search

The minimum least square error $e\ (r,\ D)$ is a distortion measure commonly used to map range block to domain block. Saupe [16] in 1995 proposed a theory that reduced the problem of finding a minimum least square. The author suggests, the integration of classification schemes proposed in [3] and fast searching method. Also, it showed that least square error $e\ (r,\ D)$ could be rewritten [16].

Here, high-encoding speed achieved at the cost of CR and the quality of the image. A linear search for the range block to domain pool mapping has computation cost $O\ (N)$, whereas the nearest neighbourhood reduced this cost to $O\ (\log N)$.

The major drawback of [16] is that it works only for the unconstrained and continuous value of scaling and luminance parameters. The nearest-neighbour approach suffers from logarithmic time requirements in the case of *k*d-tree. Also, the nearest neighbour search requires setting up the tree structure. Setting up tree structure further increases the computational cost by $O(N\ \log N)$ for N number of nodes. The performance of the algorithm decreases with an increase in the dimension of Euclidean space. Hence the size of the feature vector of range and domain block is a tradeoff for storage optimisation. Due to storage limitation, only 5–10 neighbours are considered admissible for the range block to domain pool mapping consequently mapped domain block may not be an optimal but near-optimal alternative in the neighbourhood. The article requires only a close neighbourhood search using the absolute sum of the difference technique. Parallelly, the clustering algorithm was proposed to increase the speed of the mapping of image blocks.

3.4 Clustering

A block clustering algorithm presented a solution to the time consumption problem of the local iterative function system (LIFS). A hybrid approach based on clustering and the pairwise nearest neighbour algorithm presented in 1996. The computational complexity of cluster-based FIC is less than the conventional compression algorithm. In cluster-based FIC, two-level searching is performed. Initially, a search for the best domain cluster is performed, followed by a second level search of domain block within a cluster is attempting to find an appropriate block. However, FIC algorithm using the clustering showed dependency on range block size. An increase in range block reduces the quality of cluster formed and, eventually, it leads to low image reconstruction quality. In the case of c clusters, the amount of distance calculation, n and m which is the number of range and domain blocks, respectively, is given by $nc + n(m/c)$. Optimum computational time is consumed in forming a cluster with conditions $c = \sqrt{m}$ and every cluster containing $\sqrt{m}$ some blocks. Kohonen's SOM used in [17] has the advantage of the fast creation of high-quality clusters. Also, for the fixed-sized block, significant improvement in encoding speed is noted while maintaining the image fidelity criteria. A progressive, constructive clustering algorithm suffered a slight decrease in peak signal to noise ratio (PSNR) of the image due to the choice of the best cluster. In the case of a high number of classes formed it improves the encoding speed, but eventually leads to more misclassification error.

3.5 Vector quantisation

VQ is a classic method for image compression and signal processing which had gained the attention of the researchers due to its plainness and flexibility. The encoder in VQ processes the input image as a block of pixels or vector. The encoder of VQ finds the best match against the distortion criterion from the codebook. Compression is achieved as the address is used rather than vector itself for transmission purpose. Probability density matching characteristics of VQ allow it to distinguish large density data and the data of a higher dimension. The index or address of the nearest centre of the cluster is used to represent data points. Also, range blocks encoded using VQ do not require contractivity conditions for scaling factor. The mean value of the block is utilised rather than offset for VQ encoded range blocks. The hybrid scheme of VQ and FIC reduces the complexity as compared to conventional FIC. In this scheme, the search to map domain blocks is conducted only when the centroid of a cluster is not able to approximate the block. The VQ gives a simpler decoder and fast convergence to the attractor image.

The quantisation sometimes results in the selection of sub-optimal cluster centroid. If ET is trying to reduce further, it results in lower quality of the decoded image. At a high CR, the [18] suffered from blocking effect. The Hamzaoui and Saupe [18] suggested post-processing techniques that overcome the blocking effects which are typical for encoders based on disjoint blocks.

Following approach is a hybrid scheme using convention fractal coding and transform coding. Here, the transform allows the changing of spectral content adaptively. These hybrid approaches aimed to exploit the benefit of FIC and energy compaction of the frequency domain.

3.6 Discrete cosine transforms

Most of a hybrid scheme based on FIC and DCT generates a fractal approximation of an image along with an error image. Also, the decompression of the image is comparatively easy. Barthel and Voye [19] proposed a scheme that only coded those blocks which failed to map based on the error condition. Based on [19], universal luminance transforms of higher order is proposed in the frequency domain. These hybrid approaches aimed to exploit the benefit of FIC and energy compaction of the frequency domain. DCT has the excellent advantage of applying in image compression due to its high-energy compaction properties. Scanning of DCT coefficients in the zig–zag pattern corresponds to rising spatial frequency sub-optimal root means a square distance of DCT coefficients provides a more accurate distance which makes it an efficient choice for image compression. DCT decouples the scaling and offset parameter. Hence computation required to find contractivity is easily estimated without repeatedly calculating inner product. DCT can either be applied to the entire image or on the smaller partition of the image. The latter strategy is used due to the fact a quantisation table for the large images hard to also, handle error propagation increases if DCT is applied to a large size image as compared to a small size image.

DCT-based FIC works on exploiting the benefits of the frequency domain. However, transforming the image has an added computational cost. Following an approach alternatively utilises the relationship between the local variance of range and domain block in the spatial domain to optimise the speed of FIC encoder.

3.7 Variance-based FIC

Local variance and the mean of image block are invariant of *k* self-similarity applied to the domain block. As it only changes the location of the pixel. This fact made the idea to use local variables statically useful for mapping domain block to a particular range block. The local variance of the domain block can efficiently be computed using an algorithm. A domain block of size $n \times n$ required an only $2n$ number of additions and n number of multiplications. This additional overhead of multiplication is negligible as the further calculation can be done using a lookup table. The algorithm in [20] accelerated the encoding process and slightly degraded reconstructed image quality depending on the choice of search window size. The algorithm has shown better results regarding the trade-off between domain mapping and computational complexity.

The method in [21] suffered from the poor quality of the decoded image for higher values of control parameter on the count of high CR and ET. The approach suggested in [22] has the advantage of the high-quality decoded image due to the application of neural network to sub-block mapping having high pixel variance. The sub-blocks with pixels highly correlated to the neighbourhood are coded using a general affine transform in place of the neural network. Such flexibility allowed the performance of an algorithm better than baseline variance-based fractal image coding.

Natural images contain a high correlation among the sub-blocks. It elevates the chances of faster mapping of range block from a small subset of the pool. Following the approach, exercise correlation between blocks to reduce the mapping attempt of image blocks.

3.8 Spatial correlation

Inter-block correlation exploited in [23] gave a faster solution to FIC, but the algorithm showed dependency on a threshold value opted and range size. The algorithm observed the best result in a range block of smaller size (4×4) and with the increase in block size inter-correlation reduces the performance of the algorithm also decreases. The effect of the threshold on inter-correlation leads to more computational cost. In the method [24] successfully found the local optima from correlated search space. It leads to a fewer number of bits for representing the coded image. The results of the algorithm in [25] suffered from poor reconstruction quality of the image as compared to conventional FIC.

The approaches discussed till now exercise either faster-searching method or reduction of the search space. The motive of both the strategy is to reduce the ET. However, the following categories of FIC work on breaking the quadruple loop of the encoder using the search-less approach.

3.9 Iteration-free convergence

Lepsøy *et al.* [26] proposed a modification in attractor-based image compression. The changes proposed aimed to remove the dependency on the decoder's performance on the image.

The decoder proposed in [26] required single multiplication and fewer additions compared to the conventional attractor based compression scheme. Such an algorithm could be beneficial for the fast retrieval of compressed images from the database. In [27] the authors created the same domain pool at both encoder and decoder end, which showed a reduction in distortion. The disadvantage of the algorithm is that it worked only in the spatial domain, hence results are compared only to spatial domain-based FIC algorithms. Also, LIFS suffered from blocking effects in the spatial domain which became more prominent at the lowest bit rate. This blocking effect is present regardless of the type of partitioning. Kamal [28] showed improved coding time with acceptable image quality as compared to Distasi *et al*. [29].

3.10 Local search or no search method

Bath fractal transforms to encode images proposed by [30] provided a benchmark to compare with other existing ES-based fractal image encoding algorithms. Furao and Hasegawa [31] gave a faster algorithm of no search FIC. Wang and Wang [32] used a modified grey-level transform to improve the image quality achieved through Furao and Hasegawa method. Wang *et al.* [33] proposed method provided better encoding speed and higher bit per pixel (BPP) as compared with Furao's result. No search algorithm with a fixed domain has the advantage of a higher CR as the position of the domain block is not saved in the codebook.

BFT suffered from low compression, but higher fidelity in the case of a higher order of a recursive set of mapping *vk* is considered. Furao and Hasegawa's [31] approach suffered from the poor quality of the reconstructed image. Wang and Wang's [32] solution to Furao and Hasegawa [31] low fidelity is not satisfactory as it uses only a few sub-block rather than considering blocks as a whole. The major drawback of no search algorithm is poor decoded image quality.

3.11 Parallel algorithm

Stapleton *et al.* [34] used a host and node model to explore the parallel nature of the iterated function system (IFS) calculation for image compression. The authors compared the methodology using quadtree decomposition and reposition partitioning scheme and claimed 10% usage of total time with the classifier. The approach in [34] successfully reduced the execution time, but the computation complexity of the algorithm remains O ($n4$). The authors proposed various classification schemes to reduce the search space and the number of calculations. The use of classifiers had no significant effect on communication. Further, a sub-optimal result is obtained as only a restricted search is performed. The algorithm [34] focused on low cost multicomputer for implementing parallel FIC. The architecture proposed in [35] implements full quadtree partitioning and secure higher CR. However, it suffered from the requirement of pre-processing and consuming more CPU cycles. The authors in [36] claimed better computational complexity using parallel feature extraction.

3.11 Wavelet fractal hybrid encoder

Fractal encoding based on wavelet transform using subtrees showed better results as compared to conventional FIC. Wavelet-based fractal encoding algorithm reduces the block effect present and further shows better quantisation of coding parameters. The approach in [37] failed to take advantage of the energy capability of the low-frequency sub-level band of the wavelet-transformed image. Both the methods ignored the similarity present among the sub-level of the frequency band and features of the fractal image. The subtree in the wavelet transform has the property of efficiently storing localised image features in spatial and frequency domains. It allowed, subtree to handle image energy localised in frequency and spatial domain. Also, the hybrid scheme showed faster convergence in case of choosing an overlapping subtree of the domain block. Wavelet-based encoder accelerates encoding speed as it avoids geometric transformation. It is possible to avoid the transformation as the wavelet coefficients are in the exact order as in the spatial domain. The texture in an image can be represented as fractional Brownian motions (fBM). The wavelet coefficient of fBM is stationary sequences with a self-similar covariance structure which eventually helps reduce redundancy present. The approach in [38] showed significant improvement in blocking effects in the decoded image as block partitioning performed not on the complete image instead of in an approximation sub-band. It is worth noting that a minor reduction in wavelet progressive transmission characteristic is observed. The authors also claim a 94% reduction in encoding and decoding time as compared to conventional FIC algorithms. The method proposed in [39] proved to be better to transmit images over the network with high packet loss. The authors of [40, 41] claimed a reduction in computational complexity using a wavelet-based hybrid FIC.

3.12 Genetic algorithm (GA)

GA is an excellent search technique that imitates natural selection and genetics. GA is exceptionally well suited for rough search space with more than one local solution like real-time images. Prior knowledge of the properties of the problem and coding method is required for the optimisation. High searching capacity and better resolution space are achieved with the GA evolution [24]. Another concern of GA is that if the roulette method of selection is used, the number of offspring is directly proportional to the fitness value of the parent. This relation causes premature convergence. In the case of elitism ignored in the algorithm, the number of MSE computation is equal to the product of the number of range blocks, the number of chromosomes, and pre-set iteration [42]. Further, MSE computation can be reduced using gene expression programming in convolutional neural networks [43].

Images block with better discriminative and representative information is easy to map as compared to other image blocks [44–46]. Articles in literary works [20, 47–49] claimed to improve encoding speed of FIC by using the robust features for efficient representation of image blocks. Consequently, better mapping of image sub-blocks computation cost is reduced. Lian *et al.* [22] introduced security to fractal encoding by encrypting fractal parameters. Further, in [50], a chaos-based cryptosystem is proposed to encrypt fractal parameters. Roy *et al.* [9] presented the idea of faster convergence using an approximation of the scaling parameter of the affine transform. The authors aimed for a better ET by replacing the multiplication of a matrix with the division. The new division of heuristics is bio-inspired heuristics which gained its core concepts from the social element of animals. These algorithms are derived from the living fundamentals of animals like bats, ants, and fish. Many algorithms using this class of heuristics have applied to encode the image using the fractal paradigm [6, 51]. Wu and Zang proposed a population-based metaheuristics scheme named wolf pack algorithm [52] for the optimisation problem. In [52] wolf pack algorithm is experimented with FIC to find the optimal solution to the complex FIC algorithm. The study showed the possibility of using wolf pack bioinspired for image encoding. Heuristics based algorithm's efficiency is dependent on the choice of implementation parameters.

A quick conclusion of various approaches is presented in the following table. It is observed that all the approaches work on three basic components of FIC encoder that discussed in Section 2. Each approach attempts to balance between the ET and the quality of the image. The computational cost of algorithms is influenced by their choice of search method and mapping condition. In the table, we have identified the parameters of each optimisation algorithm that affects the performance of the algorithm. Few algorithms are easily adaptable in other approaches. For example, quadtree partitioning is most adaptable to different optimisation approaches. However, wavelet and the fusion-based algorithm shows less flexibility in terms of producing a hybrid approach (Table 1).

4 Computational complexity of different approaches

Consider a greyscale image of size $N \times N$. Partition the image into non-overlapping sub-blocks of size $B \times B$ to create a range pool. Similarly, the domain pool is formed using the partitioning of overlapping sub-blocks of size twice the size of the range block. The range pool accumulates $(N/B)^2$ number of range blocks. The computational cost associated with the mapping of range and domain block is $O(B^2)$. An ES of appropriate domain-range pairs requires $O((N/4) \times (N/4) \times (N - B - 1) \times (N - B - 1))$ equals to $\sim O(N^4)$ computation. The time complexity of the algorithm is of order four and is directly proportional to the number of domain blocks in the pool. Any algorithm that attempts to reduce the ET by decreasing the size of the domain pool achieves a reduction proportion to $O(N_{\text{Domain}})$ only. Also, a smaller pool affects the quality of the decoded image.

4.1 Block classification

Various block classifications proposed to reduce the search time [54]. The scheme declines the complexity to $10 \times O((N/4) \times (N/4)) \sim O(N^2)$.

4.2 Partitioning scheme

The partitioning of an image into range and domain blocks add to the computational cost of encoding the image using FIC [55]. Partitioning of image blocks into N_{Range} and N_{Domain} blocks cost $O(k\,(N_{\text{Range}} + N_{\text{Domain}}))$ to each level.

4.3 Discrete cosine transform

DCT [11] is preferred for compression due to its energy compaction property. However, DCT itself introduces some amount of computational complexity. There are several methods for faster and efficient DCT estimation. A faster DCT takes $3N/2\,(\log N - 1) + 2$ and $N \log N + 3N/2 + 4$ number of additions and multiplications, respectively. The number of addition and multiplication are called the flop counts. Rewriting DCT in terms of discrete frequency transform (DFT) gives a faster DCT. The flop count can be reduced from $2N \log N + O(N)$ to $17/9N \log N + O(N)$ for transforming size N. *No Search*: Time complexity of region search method is $O(M \times N)$ for the constant size of the search region. A no-search or local search method reduces the time by fixing the location of an image block concerning the domain block. However, the quality of an image reduces by 25% the earlier methods.

4.4 Nearest neighbour

The elegant and innovative tree search approach reduced the ET complexity from linear to $O(\log N_{\text{Domain}})$. Although the search strategy requires additional computation to build a tree data structure to save the domain blocks. Additionally, $O(N_{\text{Domain}} \log N_{\text{Domain}})$ operations are performed before the search. In the tree structure, backtracking introduces extra $O(2k)$ constant complexity.

4.5 Parallel approach

FIC involves a large amount of independent search of domain and range block [56]. Many algorithms have proposed to enhance the encoder speed using a parallel approach. The paper [57] introduced the architecture that requires $2n + 2$ clock cycles for performing a bunch of operations. *Clustering* [58]: complexity of clustering-based FIC depends on the number of clusters, the calculation required for creating new clusters, and the number of iterations. The following is the complexity of clustering-based FIC as a function of the number of clusters (m) formed. The parameter j is the number of iterations to estimate the centre of the cluster $c(m) = I(nm + 3/2jn + jm) + mk + nm + nk/m$ parameter n and k are numbers of original and domain blocks.

The following section summaries an analysis of performance measure used in the FIC algorithms.

5 Performance measure

FIC is a computation extensive and lossy image-encoding method. The quality of the decoded image is one of the measures to analyse the performance of the algorithm. The subjective and quantitative evaluation of output determines the quality of the reconstructed image. The subjective estimation is an expensive, time-consuming, and cumbersome process. Hence quantitative measure is commonly used to evaluate the decoded image.

The study in Section 3 shows that MSE is the most straightforward and frequently used quality measure. An alternative to MSE is root means square error (RMSE). It is easy to implement quality criteria. Nearly every optimisation algorithm uses PSNR [59] to set the benchmark of the decoded image. It is a mathematically modest-quality measure for the overall image assessment. However, PSNR is unable to perceive the human

Table 1 Surveys of approaches to speedup FIC algorithm

Approach	Key parameters	Fidelity criteria	Mapping condition	Key concern and search style
block classification	quadrant variance, quadrant brightness	PSNR, BPP	RMS	the quadtree classification scheme suffers from blocking effect. Lower bits allocation further reduces the quality of the decoded image. The heuristically chosen criteria have lesser control over the fidelity. The method uses ES method to find the appropriate domain block
nearest neighbour	size of the neighbourhood	PSNR, BPP	MSE	high encoding speed is achieved at the cost of CR and quality of the image. A linear search of range to domain pool mapping required $O(N)$ computation whereas, the nearest-neighbourhood reduced this computational cost to $O(\log N)$. The nearest-neighbour search is proposed in the scheme.
partitioning scheme	shape, size of the block, number of blocks, overlapping, the shade of a block, the location of the cut	PSNR, BPP	RMS	quadtree partitioning suffers from many failed mapping attempts to map range block of considerable size. The adaptive nature of the HV partition method allows the mapping of large-sized range blocks. Also, the size of the range pool created using HV partitioning is smaller compared to the quadtree scheme. The partitioning approach supports both ES and localised or no search (NS) of the block
DCT	size of range block and domain block	PSNR, BPP, ET	MSE	DCT has the excellent advantage of applying in image compression due to its high energy compaction properties. Scanning of DCT coefficients in zig–zag pattern corresponds to rising spatial frequency. The search key is formed using such ordered coefficients. The efficiency of searching is improved as variance decreases from most to least significant key. ES, NS both exercised with DCT
DWT	wavelet decomposition level, sub-band level	PSNR, BPP, ET	MSE	the wavelet coefficients maintain their order in the spatial domain too. It is due to the energy rearrangement characteristics of the coefficient. It allows for avoiding the geometric transformation of blocks. Hence the encoding speed of the algorithm is improved
				images that are having texture details are more efficiently encoded using wavelets. Texture can be represented as fBMs. The wavelet coefficient of fBM is stationary sequences with a self-similar covariance structure which eventually helps reduce redundancy present. Wavelet-based FIC uses ES for mapping of range and domain block
Vector Quantisation (VQ)	codebook design, VQ, and its complexity	PSNR, BPP, ET, No. of clusters	RMSE	codebook generated using VQ work more efficiently for straight edges and a constant region. Fractal coder performs VQ such that it estimates a contraction map from plan to plan itself, providing a coded image which is the approximate fixed point
				the disadvantage of VQ is that it needs a fixed VQ codebook to be stored separately. Spatial transforms coding uses codebook searching
variance	window size, type of image affects the number of hit blocks	PSNR, BPP, ET	RMSE, SSIM	the algorithm in [53] accelerated the encoding process and degraded reconstructed image quality depending on the choice of search window size. The algorithm has shown better results regarding the trade-off between domain mapping and computational complexity. ES, NS, RESTRICTED SEARCH
no search	type of image, size of the range block, the threshold	PSNR, BPP, ET	RMSE	bath fractal transforms to encode images proposed by [30] provided a benchmark to compare with other existing ES-based fractal image encoding algorithms. Furao and Hasegawa [31] gave a faster algorithm of no search fractal FIC. The work in [32] used a modified grey-level transform to improve the image quality achieved through Furao and Hasegawa method. Wang *et al.* [33] proposed method provided better encoding speed and higher BPP as compared to Furao's result. No search algorithm with a fixed domain has the advantage of a higher CR as the position of the domain block is not saved in the codebook NS
spatial correlation	size of range block, search window size, the threshold value	PSNR, BPP, ET	MSE	inter-block correlation exploited in gave a faster solution to FIC, but the algorithm showed dependency on a threshold value opted and range size. The algorithm observed the best result in a range block of smaller size (4 × 4) and with the increase in block size inter-correlation reduces the performance of the algorithm also decreases. The authors noted the effect of threshold on inter-correlation that leads to more computational cost. Fast search
parallel	size of range block, threshold, partitioning	PSNR, clock frequency, CR, the threshold	SAD	preprocessing requirement. Some multiplication and division required. The platform used to implement the algorithm. ES, NS
clustering	number of clusters, number of blocks in the cluster	PSNR, ET	MSE	an increase in the number of range blocks leads to more time elapsed in searching. However, overall acceleration is achieved due to matching calculation, and variance calculation speeds up four times. ES, NS
non-iterative convergence	domain pool size	Bitrate PSNR	MSE	both encoder and decoder used a similar mean image which reduced distortion and computational complexity. Better domain pool designed using mean averaging and LBG design. Some parent/child range blocks affect the bit rate. Low rate achieved using smaller child range block. ES
GA-based FIC	mutation function, fitness function, crossover, the initial population	PSNR, BPP, the hit the block, time, speed	Mertrolpis acceptance criteria.	premature convergence: that if the roulette method of selection is used, the number of offspring is directly proportional to the fitness value of the parent. This relation causes premature convergence. ES, NS

Table 2 Effect of selection of size of range block

Size of range block	BPP	ET	CR	No. of range blocks	PSNR
2 × 2	1.0092	0.1328	7.9272	18,628	33.6847
4 × 4	0.4056	0.057	19.7227	7452	30.1061
8 × 8	0.1849	0.0224	43.2581	1888	26.8745
16 × 16	0.0853	0.0112	93.7479	1024	23.7085

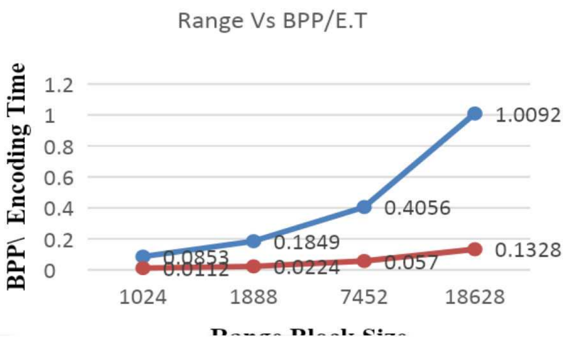

Fig. 4 *Comparison of range block size versus BPP and ET*

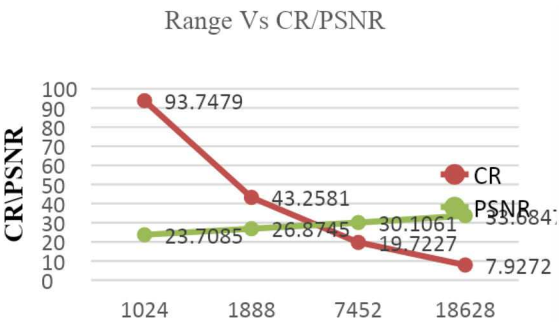

Fig. 5 *Range block size versus CR and PSNR*

visual system. Alternatively, structure similarity index (SSIM) overcomes the shortfall of PSNR.

SSIM is a direct performance measure that compares the structural information present in natural images and reconstructed images. Equation (2) defines the SSIM as a function of x and y, whereas x and y are the coordinate positions of image f.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

Parameter μ_x and μ_y are mean of reference and distorted signal, respectively. If the value of constants c_1 and c_2 is equal to zero, it produces instability in the result. Also, it is noteworthy that SSIM varieties between 0 and 1 where zero means there is no correlation between signals, and one that signifies the signal is identical.

An overall quality measure of the whole image is done using mean SSIM. Following expressions calculate mean SSIM of original image f and reconstructed image f_{decoded}.

$$\mathrm{MSSIM}(f, f_{\text{decoded}}) = 1/M \sum_{j=1}^{M} \mathrm{SSIM}(x_j, y_j) \tag{3}$$

Parameter M is a local number of windows, and x_j and y_j represent a jth window of the image. Implementation of SSIM is available online in Matlab [60].

The total time consumed in encoding and decoding is used to measure the performance of the algorithm. FIC is an asymmetrical algorithm, which takes more time to encode an image as compared to decode. Computation time can be calculated according to a particular approach in use. For example, computational time t for sub-band coding of image size $M \times N$ is calculated using [61]. The term t represents the sum of the times required for spatial contraction of domain block, measurement of self-similarity of range and domain block, and finally time for comparison of distance. Also, the time computation for sub-band coding at 2–1 resolution is given by $t^{-1} = t_{\text{LL1}} + t_{\text{LH1}} + t_{\text{HL1}} + t_{\text{HH1}}$. Here t_{LL1}, t_{LH1}, t_{HL1}, and t_{HH1} are the time required for the various sub-band. Relative acceleration is equal to the time required to encode using the full search method/time required to encode the same image using an accelerating strategy.

Table 3 Error value versus ET, CR, and PSNR

BPP	ET	CR	PSNR	Error value
0.2498	0.0338	32.0313	27.0636	1
0.2189	0.0289	36.5536	27.0425	3
0.1922	0.0234	41.6201	26.9342	6
0.1259	0.013	63.5501	24.6668	30

Finally, the CR is a widely used performance measure applied to analyse the algorithm. Compression is calculated regarding BPP. The term BPP is formulated differently for each type of algorithm [24].

Since the introduction of PIFS academicians have devoted their efforts to improve the time complexity of the fractal image algorithm. The following section takes the experimental approach to showcase the effect of fractal parameters on the performance of the algorithm. To display the trade-off between the PSNR, ET and CR graphical plots are used. One fractal parameter is kept on a common axis and the other two parameters are displayed together to show the effect of variation.

6 Experimental setup

A fractal image encoding experiment is conducted using Matlab. The 8-bit greyscale image of size 512 × 512 used to experiment. One-pixel overlapping is considered to create the domain pool. The size of the range block is double the size of the domain block. In partitioning maximum, four levels are exercised. The decoded image is obtained through a maximum of ten iterations.

7 Result

As discussed in section two, the FIC algorithm contains a quadruple loop. It causes an enormous delay in the coding of the image. As part of the study of the factors affecting the performance of baseline FIC, various parameters are varied to observe their effect on fidelity criteria. Table 2 depicts the influence of the size of the range block on CR and quality of the image. The size of the range block is varied from 2 × 2 to 16 × 16 to observe its effect on computational complexity.

It is worth observing that as the number of range blocks increases the quality of the image improves. Although more the number of blocks, ET increases and CR drop. The next two plots show the effect of image block size on the number of bits and ET (Fig. 4).

Range block is kept on the x-axis, whereas BPP and ET are shown using blue and red colours, respectively. Fig. 5 shows the effect of the number of range blocks on the CR and quality of the decoded image.

CR and PSNR are plotted on the y-axis using red and green colour, respectively. Table 3 observed the effect of the error on the performance parameters. It is noted that the high valued error threshold is more flexible in accepting the appropriate domain block. The experiment is conducted with an error threshold from 1 to 30 values.

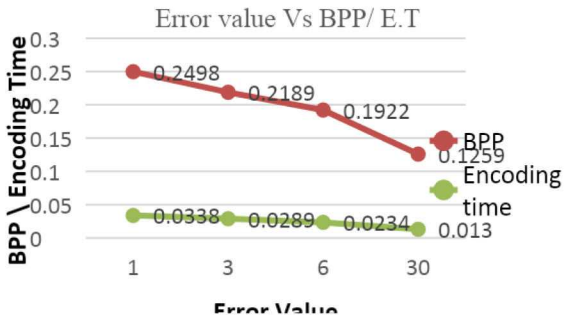

Fig. 6 *Error versus BPP and ET*

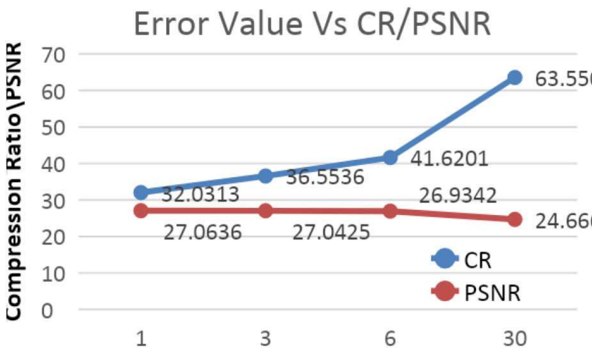

Fig. 7 *Error versus CR and PSNR*

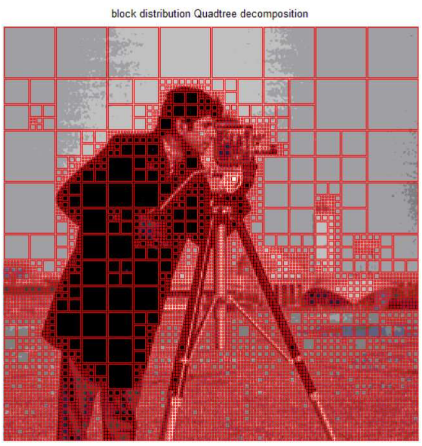

Fig. 8 *Block distribution using quadtree [size (16 × 16) error (1) sensitivity (0.1)]*

It is observed that a smaller threshold results in a lower CR and a better-quality image. A trade-off between the CR and PSNR is noted. The following graphs are created using the outcome of Table 3 (Fig. 6).

The following plot depicts the effect of raising the threshold value of the CR and PSNR. It can be observed from the graph that the CR is more affected as compared to the quality of the image by the threshold value (Fig. 7).

Block sensitivity is a fractal parameter, and its effect on the quality of the image and ET is showcased in the following Table 4. The sensitivity of the block is varied from 0.1 to 0.3. The increase in the number of mappings also rises, and the trends follow for higher values as well.

The following Table 5 is a depiction of the effect of the domain block size on the fidelity criteria. Large size domain block implies a smaller domain pool. The domain block of large size results in a more number of attempts to map to corresponding range blocks. It is worth observing that the quality of the image is unchanged by varying the size of the domain block.

Fig. 8 is a quadtree decomposition of the cameraman image.

Fig. 9 shows the mapping of the range to the domain block. Here, the domain pool is created using overlapping image blocks of size 16×16. The range pool is formed using quadtree decomposition of image blocks.

Table 4 Effect of block sensitivity on fidelity criteria

Sensitivity	ET	PSNR	No. of mapping
0.1	0.0097933	31.7614	172
0.2	0.010423	31.2767	128
0.3	0.011027	29.303	129

Table 5 Effect of size of domain block

Domain block size	Time in mapping	PSNR	No. of mapping
32	0.012753	31.7614	189
16	0.010279	31.7614	131
8	0.009603	31.7614	108
4	0.013039	31.7614	186

The mapping between domain and range block is shown using coloured lines for display purposes.

Table 6 accesses the encoder of five milestone approaches. The performance of speedup algorithms is compared using common parameters. The original Lena image of size 262 kB is used to experiment with the following table. Table 6 combines the study of different techniques applied to accelerate the encoding procedure of FIC. Various fractal parameters are studied to compare the performance of each approach. It is observed based on result of comparison that Saupe's [16] and Fisher's approach [3] alone shows lower performance as compared to hybrid approach. Three hybrid approaches in first three rows of Table 6 shows better CR and less comparisons.

The literature review delivers the responses to questions such as What the critical issues of approach are? What type of searching method is applied? What other means can be used to improve the performance? The objective of the analysis is to provide an understanding of various approaches and their strength and weakness alone.

8 Conclusion

Encoding of images is a salient aspect of data storage and management. Fractal image encoding algorithms offer a high CR, but these algorithms are computationally complex. Researchers have developed different approaches to optimise the complex calculations involved in the algorithm. Every optimisation algorithm tries to balance between the computational cost, quality of the reconstructed image and ET. The objective of this study is to provide the platform to understand the existing algorithms and identify the scope for future enhancement in the encoding.

The foundation of fractal encoding for compressing the images is discussed in Section 2. Various approaches had been proposed in the literature to enhance the efficiency of fractal encoding algorithms. Here, the study classifies the various methodology into fourteen categories. A comparative analysis of these fourteen categories with the perspective of salient parameters, advantage, disadvantage, and performance measures is discussed in Table 1. The analysis of the literature exhibits that few methodologies like block classification, GAs are more adaptable compared to others. The study also, suggests an adaptive approach performed better as compared to fixed parameter-based compression. For example, an adaptive quadtree is preferred over fixed quadtree partitioning [62]. Many algorithms are emerging that are showing improvement over conventional FIC [15, 63, 64].

The adaptive approaches take advantages of the different compatible algorithms. In the paper, the five-benchmark algorithm are observed. It is clear from the output that merging two or more approach performs better. The computational cost and adaptability of the algorithm should be kept in view before combining any algorithm. The adaptability of the algorithm depends upon the search mechanism and mapping criteria used in the encoding. The hybrid algorithm is only fruitful if it does not add to existing the computation cost. Section 4 discusses the computation cost involved with each approach. The computation cost and the

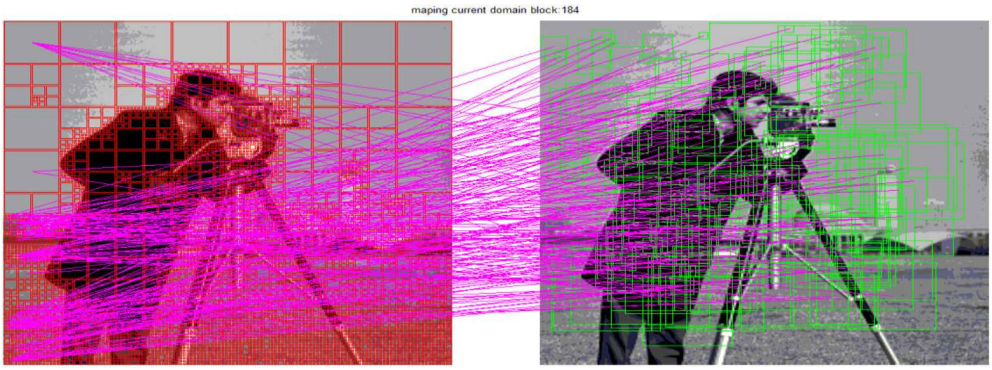

Fig. 9 *Mapping of domain block [size (16 × 16) error (1) sensitivity (0.1) number of mapping (184)]*

Table 6 Summary of various FIC algorithms

Technique	CR	BPP	No of transform	No of comparison	Transformation	Size of IFS file
Saupe–Fisher	17.2:1	0.463	4246	266,000	62.64	15,188 KB
Saupe-mass centre	14.7:1	0.542	4984	462,915	92.88	17,789 KB
Huertgen	14.73:1	0.542	4984	462,915	92.88	17,789 KB
Fisher [3]	15.88:1	0.503	4618	1,952,738	422.85	16.5 MB
Saupe [14]	16.57:1	0.482	4423	3,471,616	784.9	15,814 KB

performance of the algorithm requires a balance to achieve a practical FIC algorithm. Different performance measures that are used by various algorithms are discussed in the paper. From application to application the role of performance measure varies. FOr example, in medical imaging, the high PSNR algorithm is more relatable as compared to the CR. The scope of improving the performance of the algorithm lies with the possibility of hybrid algorithms.

We have experimentally drilled each fractal parameter to understand the performance of the algorithm.

It is observed that the partitioning scheme, the size of the block and size of search space play an essential role in the execution of the encoding algorithm. These factors influence the performance of the algorithm the most. Further, it can be concluded that the domain of the scheme is a critical factor in removing redundancy present in the image. Performance of FIC based on wavelet, parallel algorithms and genetics are yet to be explored. In future, many approaches can be brought together to increase the performance of FIC encoding.

9 References

[1] Barnsley, M.F.: 'Fractal image compression', *Notices of the AMS*, 1996, **43**, (6), pp. 657–662

[2] Jacquin, A.E.: 'Image coding based on a fractal theory of iterated contractive image transformations', *IEEE Trans. Image Process.*, 1992, **1**, (1), pp. 18–30

[3] Fisher, Y.: 'Fractal Encoding—Theory and Applications to Digital Images'

[4] Sun, Y., Xu, R., Chen, L., *et al.*: 'Image compression and encryption scheme using fractal dictionary and Julia set', *IET Image Process.*, 2015, **9**, (3), pp. 173–183

[5] Du, S., Yan, Y., Ma, Y.: 'Quantum-accelerated fractal image compression: an interdisciplinary approach', *IEEE Signal Process. Lett.*, 2014, **22**, (4), pp. 499–503

[6] Lou, L., Li, Y.: 'Research of neighborhood searching fractal image coding algorithm based on ant colony optimization'. 2015 SAI Intelligent Systems Conf. (IntelliSys), London, UK, 10th November 2015, pp. 761–764

[7] Zhao, D., Zhu, S., Wang, F.: 'Lossy hyperspectral image compression based on intra-band prediction and inter-band fractal encoding', *Comput. Electr. Eng.*, 2016, **54**, pp. 494–505

[8] Saad, A.H., Abdullah, M.Z.: 'High-speed implementation of fractal image compression in low cost FPGA', *Microprocess. Microsyst.*, 2016, **47**, pp. 429–440

[9] Roy, S.K., Kumar, S., Chanda, B., *et al.*: 'Fractal image compression using upper bound on scaling parameter', *Chaos, Solitons Fractals*, 2018, **106**, pp. 16–22

[10] Ilango, S.S., Seenivasagam, V., Madhumitha, R.: 'Hybrid two-dimensional dual tree—biorthogonal wavelet transform and discrete wavelet transform with fuzzy inference filter for robust remote sensing image compression', *Cluster Comput.*, 2019, **22**, (6), pp. 13473–13486

[11] Duh, D.J., Jeng, J.H., Chen, S.Y.: 'DCT based simple classification scheme for fractal image compression', *Image Vis. Comput.*, 2005, **23**, (13), pp. 1115–1121

[12] Jaferzadeh, K., Moon, I., Gholami, S.: 'Enhancing fractal image compression speed using local features for reducing search space', *Pattern Anal. Appl.*, 2017, **20**, (4), pp. 1119–1128

[13] Xing-Yuan, W., Dou-Dou, Z., Na, W.: 'Fractal image coding algorithm using particle swarm optimisation and hybrid quadtree partition scheme', *IET Image Process.*, 2014, **9**, (2), pp. 153–161

[14] Saad, A.M., Abdullah, M.Z., Alduais, N.A., *et al.*: 'Impact of spatial dynamic search with matching threshold strategy on fractal image compression algorithm performance: study', *IEEE Access*, 2020, **8**, pp. 52687–52699

[15] Wang, L., Liu, Z.: 'Parent block classification of fractal image coding algorithm based on 'Shizi'Feature'. Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology, Singapore, 2020, pp. 333–340

[16] Saupe, D.: 'Accelerating fractal image compression by multi-dimensional nearest neighbor search'. Proc. DCC'95 Data Compression Conf., Snowbird, UT, USA, 28th March 1995, pp. 222–231

[17] Hamzaoui, R.: 'Codebook clustering by self-organizing maps for fractal image compression', *Fractals*, 1997, **5**, (supp01), pp. 27–38

[18] Hamzaoui, R., Saupe, D.: 'Combining fractal image compression and vector quantization', *IEEE Trans. Image Process.*., 2000, **9**, (2), pp. 197–208

[19] Barthel, K.U., Voyé, T.: 'Adaptive fractal image coding in the frequency domain'. Proc. of the Int. Workshop on Image Processing, Budapest, Hungary, June 1994, Vol. 45, pp. 33–38

[20] Raj, Y.A., Alli, P.: 'Turtle edge encoding and flood fill based image compression scheme', *Cluster Comput.*, 2019, **22**, (1), pp. 361–377

[21] He, C., Yang, S.X., Huang, X.: 'Variance-based accelerating scheme for fractal image encoding', *Electron. Lett.*, 2004, **40**, (2), pp. 115–116

[22] Lian, S., Chen, X., Ye, D.: 'Secure fractal image coding based on fractal parameter encryption', *Fractals*, 2009, **17**, (2), pp. 149–160

[23] Wan, C.C., Hsieh, C.H.: 'An efficient fractal image-coding method using interblock correlation search', *IEEE Trans. Circuits Syst. Video Technol.*, 2001, **11**, (2), pp. 257–261

[24] Xing-Yuan, W., Fan-Ping, L., Shu-Guo, W.: 'Fractal image compression based on spatial correlation and hybrid genetic algorithm', *J. Vis. Commun. Image Represent.*, 2009, **20**, (8), pp. 505–510

[25] Wang, Q., Liang, D., Bi, S.: 'Fast fractal image encoding based on correlation information feature'. 2010 3rd Int. Congress on Image and Signal Processing, Yantai, People's Republic of China, 16th October 2010, Vol. 2, pp. 540–543

[26] Lepsoy, S., Oien, G.E., Ramstad, T.A.: 'Attractor image compression with a fast non-iterative decoding algorithm'. 1993 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, 27th April 1993, Vol. 5, pp. 337–340

[27] Chang, H.T., Kuo, C.J.: 'Iteration-free fractal image coding based on efficient domain pool design', *IEEE Trans. Image Process.*, 2000, **9**, (3), pp. 329–339

[28] Kamal, A.N.: 'Iteration free fractal image compression for color images using vector quantization, genetic algorithm and simulated annealing', *Turkish Online J. Sci. Technol.*, 2015, **5**, (1), pp. 39–48

[29] Distasi, R., Nappi, M., Riccio, D.: 'A range/domain approximation error-based approach for fractal image compression', *IEEE Trans. Image Process.*, 2005, **15**, (1), pp. 89–97

[30] Monro, D.M., Woolley, S.J.: 'Fractal image compression without searching. InProceedings of ICASSP'94'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Adelaide, SA, Australia, 19th April 1994, pp. V–557

[31] Furao, S., Hasegawa, O.: 'A fast no search fractal image coding method', *Signal Process., Image Commun.*, 2004, **19**, (5), pp. 393–404

[32] Wang, X.Y., Wang, S.G.: 'An improved no-search fractal image coding method based on a modified gray-level transform', *Comput. Graph.*, 2008, **32**, (4), pp. 445–450

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

[34] Stapleton, W.A., Mahmoud, W., Jackson, D.J.: 'A parallel implementation of a fractal image compression algorithm'. Proc. of 28th Southeastern Symp. on System Theory, Baton Rouge, LA, USA, 31st March 1996, pp. 332–336

[35] Jackson, D.J., Ren, H., Wu, X., *et al.*: 'A hardware architecture for real-time image compression using a searchless fractal image coding method', *J. Real-Time Image Process.*, 2007, **1**, (3), pp. 225–237

[36] Kumar, R.S., Manimegalai, P.: 'Near lossless image compression using parallel fractal texture identification', *Biomed. Signal Proc. Control*, 2020, **58**, p. 101862

[37] Davis, G.M.: 'A wavelet-based analysis of fractal image compression', *IEEE Trans. Image Process.*, 1998, **7**, (2), pp. 141–154

[38] Iano, Y., da Silva, F.S., Cruz, A.M.: 'A fast and efficient hybrid fractal-wavelet image coder', *IEEE Trans. Image Process.*, 2005, **15**, (1), pp. 98–105

[39] Yang, J.: 'Multiple description wavelet-based image coding using iterated function system', *Math. Probl. Eng.*, 2013, **2013**, pp. 1–12

[40] Sheeba, K., Rahiman, M.A.: 'Gradient based fractal image compression using Cayley table', *Measurement*, 2019, **140**, pp. 126–132

[41] Ammah, P.N., Owusu, E.: 'Robust medical image compression based on wavelet transform and vector quantization', *Inform. Med. Unlocked*, 2019, **15**, p. 100183

[42] Wu, M.S.: 'Genetic algorithm based on discrete wavelet transformation for fractal image compression', *J. Vis. Commun. Image Represent.*, 2014, **25**, (8), pp. 1835–1841

[43] Li, W., Pan, Q., Liang, S., *et al.*: 'Research on fractal image compression hybrid algorithm based on convolutional neural network and gene expression programming', *J. Algorithm. Comput. Technol.*, 2019, **13**, p. 1748302619874196

[44] Zhang, C., Zhou, Y., Zhang, Z.: 'Fast fractal image encoding based on special image features', *Tsinghua Sci. Technol.*, 2007, **12**, (1), pp. 58–62

[45] Zhou, Y.M., Zhang, C., Zhang, Z.K.: 'Fast hybrid fractal image compression using an image feature and neural network', *Chaos, Solitons Fractals*, 2008, **37**, (2), pp. 623–631

[46] Schwartz, W.R., Pedrini, H.: 'Improved fractal image compression based on robust feature descriptors', *Int. J. Image Graphics*, 2011, **11**, (4), pp. 571–587

[47] Chaurasia, V., Chaurasia, V.: 'Statistical feature extraction based technique for fast fractal image compression', *J. Vis. Commun. Image Represent.*, 2016, **41**, pp. 87–95

[48] Cao, J., Zhang, A., Shi, L.: 'Orthogonal sparse fractal coding algorithm based on image texture feature', *IET Image Process.*, 2019, **13**, (11), pp. 1872–1879

[49] Wang, J., Chen, P., Xi, B., *et al.*: 'Fast sparse fractal image compression', *PloS one*, 2017, **12**, (9), p. e0184408

[50] Ching-Hung, Y., Kwok-Wo, W.: 'Chaos-based encryption for fractal image coding', *Chin. Phys. B.*, 2012, **21**, (1), p. 010502

[51] Tseng, C.C., Hsieh, J.G., Jeng, J.H.: 'Fractal image compression using visual-based particle swarm optimization', *Image Vis. Comput.*, 2008, **26**, (8), pp. 1154–1162

[52] Wu, H.S., Zhang, F.M.: 'Wolf pack algorithm for unconstrained global optimization', *Math. Probl. Eng.*, 2014, **2014**, pp. 1–17

[53] Lee, C.K., Lee, W.K.: 'Fast fractal image block coding based on local variances', *IEEE Trans. Image Process.*, 1998, **7**, (6), pp. 888–891

[54] Wang, J., Zheng, N.: 'A novel fractal image compression scheme with block classification and sorting based on Pearson's correlation coefficient', *IEEE Trans. Image Process.*, 2013, **22**, (9), pp. 3690–3702

[55] Cardinal, J.: 'Fast fractal compression of greyscale images', *IEEE Trans. Image Process.*, 2001, **10**, (1), pp. 159–164

[56] Qureshi, K., Hussain, S.S.: 'A comparative study of parallelization strategies for fractal image compression on a cluster of workstations', *Int. J. Comput. Methods*, 2008, **5**, (3), pp. 463–482

[57] Panigrahy, M., Chakrabarti, I., Dhar, A.S.: 'Low-delay parallel architecture for fractal image compression', *Circuits Syst. Signal Process.*, 2016, **35**, (3), pp. 897–917

[58] Jaferzadeh, K., Kiani, K., Mozaffari, S.: 'Acceleration of fractal image compression using fuzzy clustering and discrete-cosine-transform-based metric', *IET Image Process.*, 2012, **6**, (7), pp. 1024–1030

[59] Wang, Q., Bi, S.: 'Prediction of the PSNR quality of decoded images in fractal image coding', *Math. Probl. Eng.*, 2016, **2016**, pp. 1–13

[60] Wang, Z., Bovik, A.C., Sheikh, H.R., *et al.*: 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process.*, 2004, **13**, (4), pp. 600–612

[61] Belloulata, K.: 'Fast fractal coding of subbands using a non-iterative block clustering', *J. Vis. Commun. Image Represent.*, 2005, **16**, (1), pp. 55–67

[62] Tong, C.S., Pi, M.: 'Fast fractal image encoding based on adaptive search', *IEEE Trans. Image Process.*, 2001, **10**, (9), pp. 1269–1277

[63] Menassel, R., Gaba, I., Titi, K.: 'Introducing BAT inspired algorithm to improve fractal image compression', *Int. J. Comput. Appl.*, 2020, **42**, (7), pp. 697–704

[64] Abedellatif, H., El-Shanawany, R., Zahran, O.F., *et al.*: 'Comparative study of wavelet transform based fractal image compression', *Menoufia J. Electron. Eng. Res.*, 2019, **28**, (ICEEM2019-Special Issue), pp. 24–28