Journal of King Saud University – Computer and Information Sciences (2018) 30, 462–469

King Saud University
Journal of King Saud University – Computer and Information Sciences
www.ksu.edu.sa
www.sciencedirect.com

Cuckoo inspired fast search algorithm for fractal image encoding

B. Mohammed Ismail [a,*], B. Eswara Reddy [b], T. Bhaskara Reddy [c]

[a] Department of Computer Science & Engineering, JNTUA Anantapuram (A.P), India
[b] Department of Computer Science & Engineering JNTUA College of Engg, Kalikiri (A.P), India
[c] Department of Computer Science & Technology, S.K.U Anantapuram (A.P), India

Received 6 August 2016; revised 12 October 2016; accepted 7 November 2016
Available online 17 November 2016

KEYWORDS
Fractal;
PSNR;
Cuckoo search;
PSO;
Genetic algorithm;
MSE

Abstract The search time and significant loss in compression are the significant constraints of the traditional fractal image compression. Hence the contemporary research contributions are aimed to discover optimal solutions to speed up the search speed with minimal loss of image significance at compression. Majority of the existing contributions achieve the search speed at the cost of decoded image quality and vice versa. In regard to this, we proposed a cuckoo inspired fast search (CIFS) technique for fractal image compression. Unlike the many of traditional models, which depend on 3 level wavelet classification, this proposed CIFS is using ordered vector of range blocks by their similarity and ordered vector of range blocks by their coordinate distance. The experimental study evinced that the proposed model is scalable and robust compared to PSO and GA based models found in contemporary literature. The significant reduction in mean square error calculations is also observed, since the only four transformations of the dihedral group are sufficient to compare for similarity here in this proposed CIFS.

1. Introduction

Fractal Image compression, which delivers a high ratio of compression and max quality at decomposing is one of the significant concepts that grabbed attention of research community since its origin in 1987. The high encoding time due to the complex search in identifying the similar blocks in given image is the significant constraint of the traditional fractal image compression, which has the iterated function system (Barnsley and Sloan, 1990) as backbone of the fractal image compression. Hence the decreasing encoding time in fractal image compression is the significant objective for many researchers. In order to do this, divergent techniques were contributed (Wu and Lin, 2010; Wang et al., 2010; Wang and Wang, 2008; Chao et al., 2007; Iano et al., 2006; Cardinal, 2001; Ismail et al., 2015).

The increase in general computing abilities of processors, the bio inspired evolutionary and artificial intelligence based solutions are considered in applications with high computations. In this context many of contributions found in contemporary literature were using evolutionary computation

* Corresponding author.
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2016.11.003
1319-1578

techniques such as genetic algorithm, fuzzy logic and Particle swarm optimization to speed up the fractal image encoding.

This paper explores a cuckoo inspired fast search technique for fractal image encoding. The objective of this work is to limit the search iterations to determine the optimal domain block for a given range block and also concentrate to reduce the mean square error computations. Unlike many of the traditional models, which are grouping the domain blocks into three classes based on their third level wavelength similarities, the proposed model is relying only on two range block vectors. First one is ordering the range blocks based on their similarities and other is ordering the range blocks by their coordinate distances. The proposed search strategy influenced by cuckoo search successfully limits the maximum number of search iterations to 6, hence reducing the overall count of mean square error computations and search time phenomenally.

The rest of the contents are organized as follows, Section 2 presents the similar contributions found in contemporary literature. Section 3 explores the detailed description and approach of the proposed model. Section 4 contains the description of experimental study and performance analysis then followed by Section 5 concluding the contributions of the work.

2. Review of related work

Wang et al. (2010) proposed a hybrid selection strategy based genetic algorithm. The optimal domain selection from two bands is evinced in hybrid select mechanism that delivered the reduction of computational time hence the encoding time reduction evinced. Wang et al. (2010), Wang and Wang (2008) devised an encoding method for fractal images based on modified gray level transform instead the search. In order to improve the PSNR of resultant decoded image, this model is adapting the fitting plane (Wang et al., 2010). The objective of the model devised in Wang et al. (2010) is achieving PSNR stability at lower and higher bits per pixel (bpp), which is achieved. Chao et al. (2007) proposed a novel search strategy that searches dissimilar domain blocks instead of similar domain block for a given range block. Iano et al. (2006) proposed a non-iterative approach, which is based on Fisher's domain classification for fractal coding. This approach uses the modified set partitioning hierarchical trees coding.

An efficient classification technique based on average and mean of pixel intensities was proposed by Jacobs et al. (1992) with a fractal image encoding technique that classifies the domain blocks based on average of pixel intensities. A similar model was devised by Wang (2010a) that uses entropy feature set to group the domain blocks as different classes. The graph based image segmentation by image content was adapted in Wang (2010b), which prepares the search space with partitions of the given input image and adaptive threshold quad-tree approach also devised to speed up the image encoding process. Kung et al. (2008) classifying the domain blocks into 4 classes of edges by the features extracted using one dimensional DCT. The similarity check was done using structure similarity index measurement (SSIM) instead mean square error (MSE), which is in the aim of reducing computational complexity. Chen et al. (2010) adapted domain block elimination strategy that discards the strictly irrelevant domain blocks at range block usage initiated. This model performs domain and range block similarity check using normalization instead of MSE and SSIM. Palazzari et al. (1999) proposed a hardware dependent fast encoding approach for fractal images. This model divides the given image into blocks and initiates sequential process on each block by an individual processor. The constraint of this approach is resource complexity, since a processor works only with related range blocks of the domain block allocated for sequential process. Another no search based fractal encoding approach is devised in Furao and Hasegawa (2004), which is similar to the models noticed in Wang et al. (2010), Wang and Wang (2008).

Zhang and Wang (2012) proposed a diamond search approach that uses wavelet transformation for domain block classification. A parallel search process using GPU (Graphics Processing Unit) was adapted for fractal image compression in Haque et al. (2014). The significant reduction in encoding time due to parallel search was observed in experimental study. Another GPU based parallel search model was proposed in Chauhan et al. (2012). This approach uses GPU cluster that divides given set of domain blocks into slave machines and triggers the range blocks for compatibility check to these slave machines in pipelined manner. This ends for range block if matching domain block is found, this process is iterative and iterates till all range blocks find the matching domain block, Wang and Zheng (2013) proposed a classification strategy that classifies domain blocks using absolute Pearson correlation coefficient in order to restrict the search to a specific class.

The existing contributions found in contemporary literature, which are briefed in the above discussion are centric to improvise the domain classification and (or) parallel process. None of the models are evincing any contribution to redefine the search process. In order to minimize the search iterations Lin and Chen (2012) proposed a PSO based search optimization with third level wavelet classification. This model is classifying the domain blocks into 3 classes, which is based on HAAR wavelet transformation. Further search is performed using Particle Swarm Optimization Technique on specific relevant class of the domain blocks. A novel dihedral group formation strategy is also adapted that limits the compatibility check of range block to 4 images of the domain block, hence the mean square error calculations are also reduced to half of the actual computations required. Similarly the search optimization was done using rank selection mechanism based genetic evolutions in Kulkarni (2015). These models achieved 177 times and 150 times reduction in search iterations and 78% and 66% reduction in *mse* calculations respectively. Though the use of PSO and GA with Rank Selection Mechanism claimed significance performance, but evinces that search reduction is probabilistic and is dependent on the selection of domain block to initiate the search. Since the search under GA and PSO is usually initiated at random domain block, which could be directed to least optimistic search path.

In contrast to these models, our proposal is cuckoo inspired search, which is with a specific number of search iterations as compared to traditional cuckoo search (Yang and Deb, 2009) with maximum iterations of 15. Our search model is limited to 1–6 search iterations where 1 in best case and 6 in worst case. The existing models are classifying the domain blocks but the proposed model is ordering the range block by their similarity and coordinate distance, this finds minimal *mse* computations with 96% reduction and search iterations reduced by 224 times when compared with full search.

3. Fractal image encoding by cuckoo inspired fast search

Fractal image compression performs the encoding by the following two properties

(1) Property of local self-similarity property.
(2) Partitioned Iterative Function System.

The exploration of the traditional fractal image compression under these properties is explained with an example in Section 3.1.

3.1. Fractal image compression

Let f be the given gray scale image of square with length and breadth as m (let say m is 128), Then partition the image into square blocks of size $h \times h$ (in this example h is 8), which are referred as domain blocks. The total number of overlapping domain blocks from the given image f is 14,661 (which can be estimated as $(m - h + 1)^2$). Partition the given image into non-overlapping blocks called range blocks of size n (here in this example n is 4) such that $2 \times n = h$. The total number of range blocks of given image f is 1024 (which can be estimated by $(m/n)^2$) blocks. In order to find the fractal transformation of each range block rb, the most compatible domain block db will be traced from the pool of domain blocks, the compatibility check of a range block and domain block demands the size of both blocks be same, hence the all domain blocks will be reduced to the size $n \times n$.

Danahy et al. (2007) devised a technique called logical transformation based image resizing, which is used here in this contribution to resize the domain blocks to the size reflected by range blocks. The description of the model called logical transformation that adapted to resize the domain blocks into the size of range blocks follows.

The logical system transform converts binary data into a sum of primary implicates representation. The logical transform, is useful for quickly determining the Boolean minimized form (vector y) from an input signal. Unlike other common transforms, the inverse operation of the logical transform is not a mirror image of the forward transform. Instead, the original binary data are generated from the sum of primary implicates through a process called implicate expansion.

In order to do this the search for compatible domain block visits all domain blocks available. The mean square error (mse) observed between a range block rb and a domain block db are used to estimate their similarity. Less mse indicates the more similarity between the given blocks. The mean square error between rb and db can be calculated by using Eq. (1).

$$mse_{db \to rb} = \frac{1}{n \times n} \sum_{dgi=1}^{|DG|} (db_{dgi} - rb_{dgi})^2 \tag{1}$$

Here in the Eq. (1), $n \times n$ is the size of the range block, dgi is the index of the image in dihedral group, $|DG|$ is the size of the dihedral group (which is 8), db_{dgi} is the dihedral group image of the domain block db at index dgi and rb_{dgi} is the dihedral group image of the range block at index dgi.

In order to check the fractal similarity between domain and range blocks, the dihedral group (of 8 orientations) of images will be formed from the domain block. The dihedral group contains 8 orientations and they are

(1) Original domain block.
(2) Domain block db flipped horizontally (in 90^0) that forms db_h.
(3) Domain block db flipped vertically (180^0) that forms db_v.
(4) Domain block db flipped horizontally and vertically (270^0)that forms db_{hv}.
(5) Domain block db flipped on diagonal line ($x = y$) that forms db_d.
(6) Diagonally flipped domain block db_d flipped again horizontally (in 90^0) that forms db_{dh}.
(7) Diagonally flipped domain block db_d flipped again vertically (in 180^0) that forms db_{dv}.
(8) Diagonally flipped domain block db_d flipped again horizontally and vertically (in 270^0) that forms db_{dhv}.

Hence the total number of domain block images to be compared to range block in order to find the compatibility are $(m - h + 1)^2 \times 8$ (results 117128 for given example). Here 8 indicates the total number of images in the dihedral group of each domain block and $(m - h + 1)^2$ indicates the total number of domain blocks. This confirms that the total number of mse calculations required for each range block for the given example is 117128 and total number of mse calculations for encoding the given image f is 119939072 (which is estimated as 117128×1024). Hence it is clear that search for a compatible domain block for given range block is computationally complex that further escalates the process complexity of fractal encoding.

3.2. Cuckoo search

Cuckoo search is a search technique stimulated by the holoparasite act of some cuckoo birds. The species of type cuckoo is unable to complete its reproduction cycle without proper host (nest of the other type of birds that contains eggs resemble to cuckoo bird egg). The Cuckoo bird places its egg (s) in the nest of host. The strategy of search followed by a cuckoo bird is adapted in numerous fields (Qin et al., 2014; Mohapatra et al., 2015; Sun et al., 2016). Cuckoo search executes under three traditional rules (Yang and Deb, 2010) and they are

(i) Cuckoo selects a host nest randomly to place the egg.
(ii) The nest contains most compatible eggs that compared to cuckoo egg enables the reproduction of the cuckoo.
(iii) The finite number nests (usually the 15) Yang and Deb, 2009 are adapted for cuckoo search.

The probability factor to notify the cuckoo egg as artifact by a host bird is $\{P(a) \exists a \in (0, 1)\}$. In order to optimize the initial nest of the search, search follows the techniques such as Levy flights and random walks (Yang and Deb, 2009).

3.3. Nest formation

Each nest is representing a domain block db such that dihedral transformations of that block as eggs. The fractal transformation allows the eight Dihedral transformations (which are considered as the eggs placed in the nest represented by respective domain block db) as shown in Table 1.The index $\{h, v, hv, dh, dv, dhv\}$ represents the respective dihedral transformation of the domain block db. The transformation of the db can be as flip the db along horizontal db_h, vertical db_v and both horizontal and vertical db_{hv}, further flip the db on main diagonal line as db_d then flip db_d horizontal db_{dh}, vertical db_{dv} and both horizontal and vertical db_{dhv}. In order to transform the domain block, the center of the domain block is considered as coordinate origin. The final group of dihedral transformations of domain block db is {"domain block (db)", "horizontal flip (db_h) ", "vertical flip (db_v)", "horizontal and vertical flip (db_{hv})-", "diagonal flip (db_d)", "diagonal and horizontal flip (db_{dh})", "diagonal and vertical flip (db_{dv})", "diagonal, horizontal and vertical flip (db_{dhv})"} ($\{db, db_h, db_v, db_{hv}, db_d, db_{dh}, db_{dv}, db_{dhv}\}$).

3.4. Nest search

The nest search is redefined such that the search limited to fixed number of nests. In order to this, the levy flights (Yang and Deb, 2009) technique is adopted. The metrics used in this levy flight technique to initiate the search are:

Ordered range blocks by the dihedral transform similarity

(i) Order the range blocks by their pixel coordinates distance appearing in original image, which enables each range block to begin the search for compatible nest from the nests of those fixed for its neighbor range block.
(ii) Order the range blocks by their pixel coordinates distance appearing in original image that initiates search from the nest fixed for the neighbor nest under coordinate distance, if compatible nest is not found under first levy factor.

Hence the range blocks will be order in two dimensions called (i) by their similarity and (ii) by their coordinate distance. Here, the center of the each range block is considered as coordinate

3.4.1. Ordering of the range blocks

In order to simplify the search for appropriate domain block for each range block, the range blocks will be ordered by their similarity. In regard to this each range block follows the other range block that contains minimal Mean Square Error. This practice helps to identify the compatible domain block by one iteration of search under the transitive property (Bianchi, 2000), which is as follows.

$$\text{If } (mse_{db\to rb_i} \equiv 0) \wedge (mse_{rb_i \to rb_{i+1}} \equiv 0) \text{ then } mse_{db\to rb_{i+1}} \equiv 0 \qquad (2)$$

Table 1 Fractal transformations of the dihedral group.

db: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	db_h: $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	db_v: $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	db_{hv}: $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
db_d: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	db_{dh}: $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	db_{dv}: $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	db_{dhv}: $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$

Here in the Eq. (2), $mse_{db\to rb_i}$ is the mean square error of the domain block db and range block rb_i exists at ith range block of the range blocks ordered by their similarity. Similarly $mse_{rb_i\to rb_{i+1}}$ is the mean square error of the range blocks rb_i and rb_{i+1} exists in sequence of positions i and $i+1$ in range blocks ordered by their similarity.

If $mse_{rb_i\to rb_{i+1}} \neq 0$, then search continues on other factor of the levy flights .In order to this, the range blocks will be ordered such that range block rb_i is followed by other range block rb_{i+1}, if the distance between rb_i and $rb_{i+!}$ is minimal than the distance between rb_i and other range block. This factor enables to initiate search at a nest represented by a domain block db that contains the coordinate of the range block, further the search can limit to the nest represented by db and its neighbor nests (Je and Park, 2013).Nests concerned to the domain blocks of top, bottom, left and right represents the top, bottom, left and right portions in image. Hence the overall nest search is limited to ($\leqslant$6).

Further the nest compatibility check will be performed, which is as follows:

The respective domain block of the nest selected for compatibility check, all of the eight Dihedral transformations are performed and the best Dihedral index along with corresponding contrast offset c, brightness offset b, and MSEs respective to the range block involved in search process can be obtained. If any of the MSEs is found to be nullified, then the search will be stopped and range block placed in the nest obtained, else continues search under the factors considered for levy flights. The offsets of contrast c, brightness b can be computed as follows.

$$c = \frac{n^2\langle db_o, rb\rangle - \langle db_o, \bar{v}\rangle\langle rb, \bar{v}\rangle}{n^2\langle db_o, db_o\rangle - \langle db_o, \bar{v}\rangle}, \quad b = \frac{\langle rb, \bar{v}\rangle - c\langle db_o, \bar{v}\rangle}{P_{rb}} \qquad (3)$$

Here in this equation Eq. (3) P_{rb} represents the total number of pixels found in range block and $\bar{v}$ is the 8×8 matrix, such that 1 is the value of all entries, and $\langle .\, .\rangle$ is the scalar product of the matrices (multiple vectors) that accentuating the geometric significance. The sub-sampled domain block that is denoted by db_o, the o represents the index $\{db, db_h, db_v, db_{hv}, db_d, db_{dh}, db_{dv}, db_{dhv}\}$ of optimal dihedral group image of db.

The search criterion of levy factors limits the compatibility check to specific number of nests, which is at most to 6 iterations. Nest nbn is fixed for neighbor block under similarity; nest $nebn$ represented by domain block contains the coordinate of the input range block or the nests represented by the neighbor domain blocks ($\leqslant$4) of the domain block representing $nebn$. The MSE calculation is also limited to four dihedral images, which are shown in Table 2.

4. Experimental study and performance analysis

Execution of proposed algorithm is carried out on Core i5-5200U; 4 GB RAM 1 TB Hard-Disk, 2 GB Graphics with Windows 10. The implementation of proposed method is carried out on NVIDIA Ge Force GTX 480 GPU Using CUDA language. Experimental results are carried out on 3 standard test images.

The similarity check between each range block and all 8 dihedral transformations of domain block given in Table 2 should be performed in order to find the domain block that

Table 2 Relations observed between $F_{[10]}$ and $F_{[0\,1]}$ at dihedral transformation.

	db	db_h	db_v	db_{hv}	db_d	db_{dh}	db_{dv}	db_{dhv}
Third level high and low frequency bands ($HL3$)	$F_{+[1\,0]}$	$F_{+[1\,0]}$	$F_{-[1\,0]}$	$F_{-[1\,0]}$	$F_{+[0\,1]}$	$F_{+[0\,1]}$	$F_{-[0\,1]}$	$F_{-[0\,1]}$
Third level low and low frequency bands ($LL3$)	$F_{+[0\,1]}$	$F_{-[0\,1]}$	$F_{+[0\,1]}$	$F_{-[0\,1]}$	$F_{+[1\,0]}$	$F_{-[1\,0]}$	$F_{+[1\,0]}$	$F_{-[1\,0]}$

Table 3 Input factors obtained from given image.

M (image size)	256
N (Range block size)	8
R (Number of range blocks	1024
D (Number of domain blocks)	58081

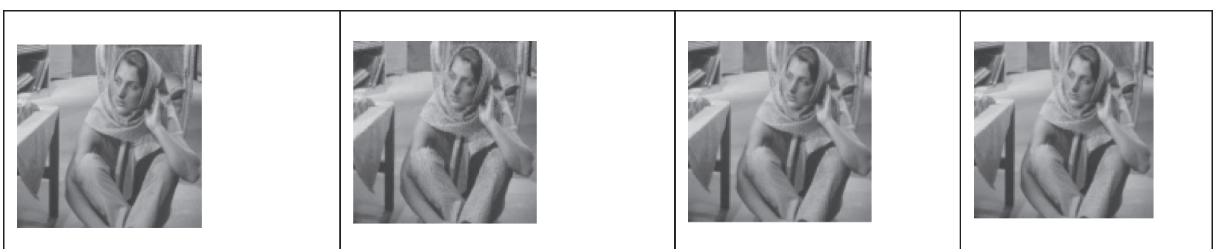
reflects optimal similarity with given range block. The results $F_{[1\,0]}$ and $F_{[0\,1]}$ of Dihedral transformations explored Table 2, which are indicating that $F_{\pm[1\,0]}$ and $F_{\pm[0\,1]}$ are the same. Hence the *mse* calculations can be reduced to 4 that optimizes the encoding time.

The edge directions represented by dihedral transformation set $\{db, db_h, db_v, db_{hv}\}$ and $\{db_d, db_{dh}, db_{dv}, db_{dhv}\}$ are identical, which is due to the similarity between $|F_{[1\,0]}|$ and $|F_{[0\,1]}|$ Lin and Chen, 2012. Hence the *mse* computations can be limited to 4. If both *db* and *rb* belongs same region then the *mse* calculation limits to $\{db, db_h, db_v, db_{hv}\}$, else *mse* calculations are required only on $\{db_d, \mathrm{db}_{dh}, db_{dv}, db_{dhv}\}$ (Kulkarni, 2015). Therefore the *mse* computations restricted to 4 and total *mse* calculations are reduced to the half of the computation count observed in full search.

4.1. Encoding

Upon discovery of the compatible nest, together with the corresponding contrast offset c, brightness offset b, position (x, y) of the respective domain block *db* of the nest encountered and compatible dihedral image $\{db||db_h||db_v||db_{hv}||db_d||db_{dh}||db_{dv}||db_{dhv}\}$ of the *db* constitutes the fractal code of the given range block *rb*.

Table 3 shows the input factors obtained from the test images Barbara, Lena and Pepper all of size (512×512). Figs. 1–3 shows original images Barbara, Lena, Pepper and their decoded images by CIFS and other benchmarking models called PSO with Wavelet Classification (Lin and Chen, 2012) and GA with Ranking Selection Mechanism (Kulkarni, 2015).

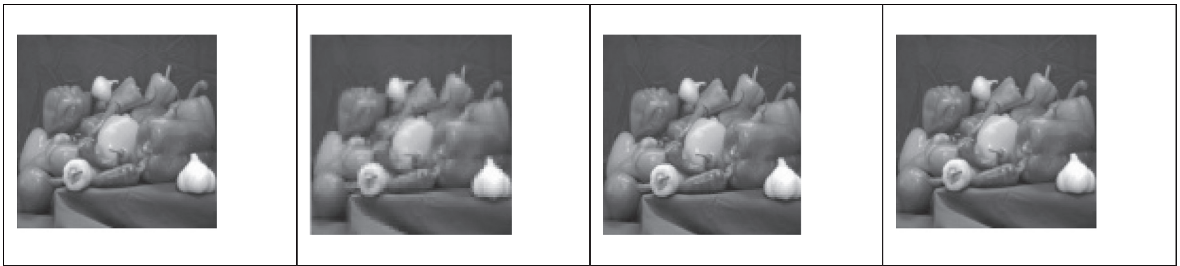


(a) Original | (b) Decoded by PSO with Wavelet Classification | (c) Decoded By GA with Ranking Selection Mechanism | (d) Decoded by CIFS

Figure 1 The original and decoded Barbara images of algorithms adapted.

(a) Original | (b) Decoded by PSO with Wavelet Classification | (c) Decoded By GA with Ranking Selection Mechanism | (d) Decoded by CIFS

Figure 2 The original and decoded Lena images of algorithms adapted.



(a) Original

(b) Decoded by PSO with Wavelet Classification

(c) Decoded By GA with Ranking Selection Mechanism

(d) Decoded by CIFS

Figure 3 The original and decoded Pepper images of algorithms adapted.

Table 4 Performance metrics and the results obtained from full search, PSO with wavelet classification (Lin and Chen, 2012), GA with RSM (Kulkarni, 2015) and CIFS.

	Full search	PSO with wavelet classifier (Lin and Chen, 2012)	GA with rank selection mechanism (Kulkarni, 2015)	Cuckoo inspired fast search
Max number of MSE calculations	475799552	2673031	3171997	2119680
Max search iterations per range block	58081	326	387	258
Search optimization (times of reduction compared to full search	0	177	150	224
PSNR (in decibels)	28.91	27.404	27.643	28.001
CPU time in sec (C++ program)	1433.281	8.344	9.5552067	6.3701378
GPU time in Sec (CUDA program)	102.37721	0.596	0.6825148	0.4550098

The metrics listed in Table 4 were compared for PSO with wavelet Classifier, GA with RSM and proposed CIFS, which are evincing the advantage of CIFS over other two models (Lin and Chen, 2012; Kulkarni, 2015).The maximal number of search iterations for each range block (in the case of a non-fractal image) are 326 (which is 177 times less than the search iterations observed for full search), 387 (which is 150 times less than the full search) and 258 in respective order of PSO with Wavelet Classification (Lin and Chen, 2012), GA with RSM (Kulkarni, 2015) and proposed model called CIFS shown in Fig 4.

Since the compression process is lossy, the decoded image quality is estimated by Peak Signal Noise Ratio (PSNR), which is calculated as shown in Eq. (4).

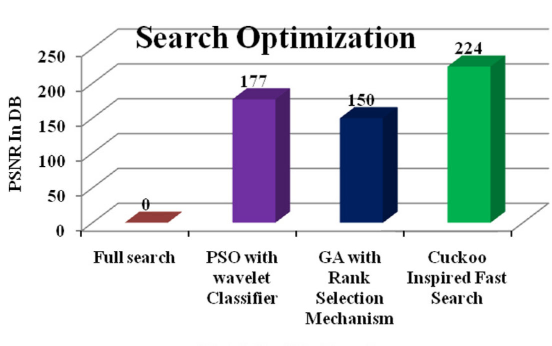


Figure 4 Comparison of Search optimization against full search.

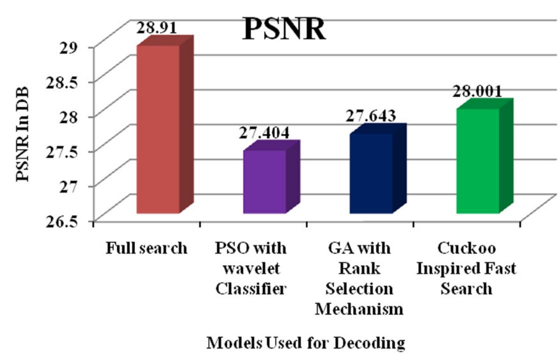


Figure 5 Comparison of PSNR observed for proposed CIFS and models adapted from Lin and Chen (2012) and Kulkarni (2015).

$$PSNR = 10\log_{10}\left(\frac{255 \times 255}{mse_{oi\rightarrow di}}\right) \quad (4)$$

Here in the Eq. (4) $mse_{oi\rightarrow di}$ is the mean square error observed between original image *oi* and decoded image *di*. The PSNR observed from decoded images are 27.404, 27.643, 28.001 in respective order of models in Lin and Chen (2012), Kulkarni (2015) and proposed CIFS. These PSNR values are evincing that the proposed model CIFS out performs as compared to other two models (Lin and Chen, 2012; Kulkarni, 2015) (Fig. 5). GA with RSM retains decoded picture quality compared to PSO with wavelet classification, whereas PSO with wavelet classification reduces search itera-

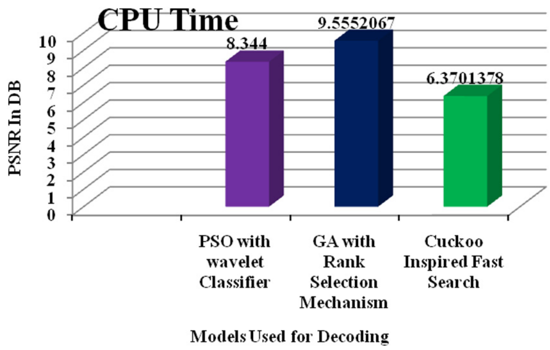


Figure 6 Completion time observed on CPU for PSO with Wavelet Classification, GA with RSM and CIFS.

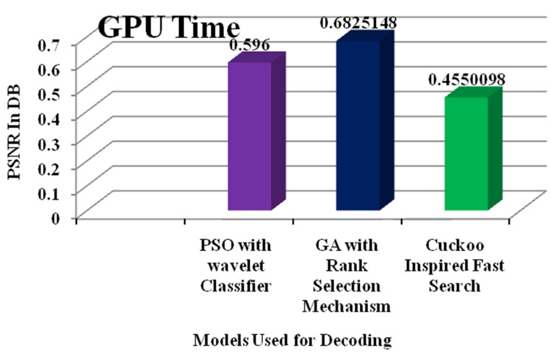


Figure 7 Completion time observed on GPU for PSO with Wavelet Classification, GA with RSM and CIFS.

tions and MSE calculations compared to GA with RSM. The proposed CIFS is found to be best with minimal search iterations, minimal MSE calculations and maximal PSNR compared to other two models.

The search completion time observed in respective order of full search, PSO with Wavelet classification, GA with RSM and proposed CIFS for Central Processing Unit (implemented by C + +) are 1433.281, 8.344, 9.5552067 and 6.3701378s, for Graphics Processing Unit (GPU) (implemented by CUDA) are 102.37721, 0.596, 0.6825148 and 0.4550098s. The values observed for the metric called search completion time clearly evince that CIFS search is much faster compared to other two models as shown in Figs. 6 and 7

5. Conclusion

This paper devises a cuckoo inspired fast search (CIFS) technique to accelerate the encoding process of a fractal image. The CIFS is using vectors of range blocks that are ordered by their similarity and coordinate distance respectively. The Cuckoo search is modified such that search performs only on limited nests (maximum six) and initial nest selection for search process is done by levy flights strategy. The factors used for the levy flights strategy are (i) The nest identified for the predecessor range block of the vector (that ordered by similarity) may be compatible or (ii) The nest represented by the domain block that contains the coordinates of the range block or nests represented by its neighbor blocks are compatible. The overall MSE calculation reduces to 224 times compared to full search. The completion time under CPU and GPU are also assessed, which are evincing that CIFS completion time is best and robust. The performance analysis of the CIFS is done by comparing with other contributions called PSO with Wavelet Classification (Lin and Chen, 2012), GA with RSM (Kulkarni, 2015) found in contemporary literature. The CIFS has achieved optimal speed (224 times faster than the full search) compared to PSO with Wavelet Classification (177 times faster than the full search) and GA with RSM (150 times faster than the full search). The CIFS completion time on CPU (6.3701378s) and GPU (0.4550098s) is significantly lesser than the respective completion times on CPU (8.344) and GPU (0.596) implementation of PSO with Wavelet Classification and respective completion times on CPU (9.5552067) and GPU (0.6825148) implementation of GA with RSM. The PSNR observed for respective decoded images of full search, PSO with Wavelet Classification, GA with RSM and proposed CIFS are 28.91, 27.404, 27.643 and 28.001 decibels respectively. These results evince that the PSNR obtained for proposed CIFS is near (with the difference of 0.909) to the PSNR observed for full search model, which is minimal compared to other two models (Lin and Chen, 2012; Kulkarni, 2015) considered. This work can be extended further to minimize the number of MSE calculations occurred to order the range blocks by their similarity, which will be the future direction of our research.

References

Barnsley, Michael F., Sloan, Alan D., 1990. Methods and apparatus for image compression by iterated function system. U.S. Patent No. 4,941,193.

Bianchi, Mariagrazia et al, 2000. On finite solvable groups in which normality is a transitive relation. J. Group Theory 3 (2), 147–156.

Cardinal, Jean, 2001. Fast fractal compression of greyscale images. IEEE Trans. Image Process. 10 (1), 159–163.

Chao, Zhang, Yiming, Zhou, Zengke, Zhang, 2007. Fast fractal image encoding based on special image features. Tsinghua Sci. Technol. 12 (1), 58–62. ISSN 1007-0214.

Chauhan, Munesh Singh, Negi, Ashish, Rana, Prashant Singh, 2012. Fractal image compression using dynamically pipelined GPUclusters, Proceedings of the Second InternationalConference on Soft Computing for Problem Solving (SocProS 2012), pp. 575–581.

Chen, Hsiu-Niang, Chung, Kuo-Liang, Hung, Jian-Er, 2010. Novel fractal image encoding algorithm using normalized one-norm and kick-out condition. J. Image Vis. Comput. 28 (3), 518–525. Elsevier.

Danahy, Ethan E., Agaian, Sos S., Panetta, Karen A., 2007. Algorithms for the resizing of binary and grayscale images using a logical transform. Electronic Imaging 2007. International Society for Optics and Photonics.

Furao, Shen, Hasegawa, Osamu, 2004. A fast no search fractal image coding method. J. Signal Process. Image Commun. 19 (5), 393–404. Elsevier.

Haque, M.D. Enamul, Al Kaisan, Abdullah, Saniat, Mahmudur R., Rahman, Aminur, 2014. GPU accelerated fractal image compression for medical imaging in parallel computing platform. J. CORR abs/1404.0774.

Iano, Yuzo, da Silva, Fernando, Cruz, Ana, 2006. A fast and efficient hybrid fractal wavelet image coder. IEEE Trans. Image Process. 15 (1), 98–105.

Ismail, M., Basha, B.M., Eswar Reddy, B., 2015. Improved fractal image compression using range block size. In: IEEE Proc International Conference on Computer Graphics, Vision and Information Security (CGVIS). 284, 289.

Jacobs, E.W., Fisher, Yuval., Boss, R.D., 1992. Image compression: a study of the iterated transform method. J. Signal Process. 29 (3), 251–263. Elsevier.

Je, Changsoo, Park, Hyung-Min, 2013. Optimized hierarchical block matching for fast and accurate image registration. Signal Process. Image Commun. 28 (7), 779–791.

Kulkarni, Amita N., et al., 2015. Fractal image compression using genetic algorithm with ranking select mechanism. Communication, Information & Computing Technology (ICCICT), 2015 International Conference on. IEEE.

Kung, C.M, Yang, W.S., Ku, C.C., Wang, C.Y., 2008. Fast fractal image compression base on block property. In: IEEE 2008 International Conference on Advanced Computer Theory and Engineering, Phuket, DEC, pp. 477–481.

Lin, Yih-Lon, Chen, Wen-Lin, 2012. Fast search strategies for fractal image compression. J. Inf. Sci. Eng. 28 (1), 17–30.

Mohapatra, P., Chakravarty, S., Dash, P.K., 2015. An improved cuckoo search based extreme learning machine for medical data classification. Swarm Evol. Comput. 24, 25–49.

Palazzari, Paolo, Coli, Moreno, Lulli, Guglielmo, 1999. Massively parallel processing approach to fractal image compression with near-optimal coefficient quantization. J. Syst. Arch. 45 (10), 765–779. Elsevier.

Qin, S., Liu, F., Wang, J., Sun, B., 2014. Analysis and forecasting of the particulate matter (PM) concentration levels over four major cities of China using hybrid models. Atmos. Environ. 98, 665–675. http://dx.doi.org/10.1016/j.atmosenv.2014.09.046.

Sun, X., Sun, W., Wang, J., Zhang, Y., Gao, Y., 2016. Using a Grey–Markov model optimized by Cuckoo search algorithm to forecast the annual foreign tourist arrivals to China. Tour. Manage. 52, 369–379.

Wang, Hai, 2010a. Fast image fractal compression with graph- based image segmentation algorithm. Int. J. Graph. 1 (1), 19–28.

Wang, Hai, 2010b. Fast image fractal compression with graph-based image segmentation algorithm". Int. J. Graph. 1 (1), 19–28.

Wang, Xing-Yuan, Wang, Shu-Gua, 2008. An improved no-search fractal image coding method based on a modified gray-level transform. Comput. Graph. 32, 445–450. Elsevier.

Wang, Jianji, Zheng, Nanning, 2013. A novel fractal image compression scheme with block classification and sorting based on Pearson's correlation coefficient. IEEE Trans. Image Process. 22 (9), 3690–3702. Proc.IEEE.

Wang, Xing-Yuan, Wang, Yuan-Xing, Yun, Jiao-Jiao, 2010. An improved no search fractal coding method based on fitting plane. Image Vis. Comput. 28, 1303–1308. Elsevier.

Wu, Ming-Sheng, Lin, Yih-Lon, 2010. Genetic algorithm with hybrid select mechanism for fractal image compression. Dig. Signal Process. 20, 1150–1161. Elsevier.

Yang, X.S., Deb, S., 2009. Cuckoo search via levy flights. Nat. Biol. Inspired Comput., 210–214 NaBIC 2009. World Congr., 2009.

Yang, X.S., Deb, S., 2010. Engineering optimisation by cuckoo search. Int. J. Math. Model. Numer. Optim. 1, 330–343.

Zhang, Yi, Wang, Xingyuan, 2012. Fractal compression coding based on wavelet transform with diamond search, journal of. Nonlinear Anal. Real World Appl. 13 (1), 106–112. Elsevier.