

RESEARCH

Open Access

# Intelligent fuzzy approach for fast fractal image compression

Ali Nodehi<sup>1</sup>, Ghazali Sulong<sup>1</sup>, Mznah Al-Rodhaan<sup>2</sup>, Abdullah Al-Dhelaan<sup>2</sup>, Amjad Rehman<sup>3</sup> and Tanzila Saba<sup>4\*</sup>

## Abstract

Fractal image compression (FIC) is recognized as a NP-hard problem, and it suffers from a high number of mean square error (MSE) computations. In this paper, a two-phase algorithm was proposed to reduce the MSE computation of FIC. In the first phase, based on edge property, range and domains are arranged. In the second one, imperialist competitive algorithm (ICA) is used according to the classified blocks. For maintaining the quality of the retrieved image and accelerating algorithm operation, we divided the solutions into two groups: developed countries and undeveloped countries. Simulations were carried out to evaluate the performance of the developed approach. Promising results thus achieved exhibit performance better than genetic algorithm (GA)-based and Full-search algorithms in terms of decreasing the number of MSE computations. The number of MSE computations was reduced by the proposed algorithm for 463 times faster compared to the Full-search algorithm, although the retrieved image quality did not have a considerable change.

**Keywords:** Fractal image compression; Optimization method; Image quality; Intelligent competitive algorithm; Imperialist competitive algorithm; Genetic algorithms

## 1 Introduction

Image compression is the current mainstream research topic in image processing [1,2] and is still a viable research area, owing to the growing need for multimedia data transmission and storage [3]. Fractal image compression (FIC) is a lossy compression method developed in the late 1980s by Barnsley and Lurd [4]. Jacquin [5] presented an automatic algorithm called baseline fractal image compression. In [6,7], classification methods for determining the domain blocks were proposed. In [8], the information existing within the matched range block (a neighborhood matching method) is used for the encoding process, but is based on the spatial correlation between the range and the domain block. Truong and Jeng [9] proposed the use of the discrete cosine transform (DCT) inner product-based algorithm for removing the redundancy that exists while the eight mapping of the domain block is computed. Furaó and Hasegawa [10] introduced a no-search encoding fractal algorithm for faster encoding process by choosing a domain block

with the same center as the range block; in essence, the domain block is encoded as a matched block.

In recent years, some literature combined the genetic algorithm (GA) with the FIC method to address the encoding time problem. In [11-17], the chromosome fitness for the evolutionary process was determined by calculating the distance from domain to range blocks. In [18], the use of Fisher classification, the genetic compression algorithm, and quad-tree partitioning were adopted for faster encoding time. In their work, the avoidance of premature convergence of GA was taken into consideration as one of the ways to minimize computation time. A spatial correlation GA was introduced in [19] for speeding up FIC. Their approach were of two stages, performing the spatial correlation in image for both range block and domain block so as to obtain local optima, but if the local optima was not the actual optima, a self-similarity approach is adopted to determine the best match within the image blocks. In [20], schema genetic algorithm was proposed; in their approach, the genetic operators based on the schema theorem within the evolutionary process are used to accelerate the encoding process and also conserve the image quality. In [21], classification of image blocks based on DCT coefficients was applied with GA

\* Correspondence: tsaba@pscw.psu.edu.sa

<sup>4</sup>College of Computer and Information Sciences, Prince Sultan University, Riyadh 12435, Kingdom of Saudi Arabia

Full list of author information is available at the end of the article

to minimize encoding time. In [22], Similar Extension algorithms as well as an algorithm for partitioning the images by means of irregular region in the FIC have been introduced, which were purposed for constructing the irregular region transformation. They increased considerably the compression ratios that were obtained by conventional block-based partition. Based on local fractal dimensions, dynamic classification scheme is presented in [23]. In [24], a FIC scheme was introduced based on the fact that the affine similarity between two blocks in FIC is equivalent to the absolute value of Pearson's correlation coefficient between them. Despite all the proposed methods for encoding time reduction discussed here, the problem of encoding time still exists, especially for natural scene images. Imperialist competitive algorithm (ICA) is an optimization algorithm introduced through socio-political process of imperialistic competition that was proposed in 2008 [25]. The aforementioned process is responsible for handling various optimization tasks. In different domains, several characteristics of ICA such as effectiveness, limitedness, and applicability have been investigated. Improvements in both global optima and convergence rate are the major achievements of ICA. This paper aims at speeding up fractal encoding through an ICA. The proposed algorithm is composed of two steps: First, according to the image blocks' edge property, classify all image blocks using the DCT coefficients. Second, using results obtained from the first step, the algorithm divides the countries into two groups - developed countries and undeveloped countries. For reducing the mean square error (MSE) computations, we calculate just the MSE value.

The paper is composed of four further sections. Section 2 introduces the basic theory behind FIC. In Section 3, the concept behind the ICA is discussed, while in Section 4, the implementation of the fractal encoding strategy using the proposed ICA is presented. The experimental results of the proposed approach are presented in Section 5. Finally, the conclusion is drawn and possible future of the research is presented.

## 2 Fractal image compression

Fractal image compression (FIC) can be practically implemented if we assume that  $f$  is a given  $256 \times 256$  gray level image. The range pool  $R$  is considered as the set of all non-overlapping blocks of size  $8 \times 8$ , making up 1,024 blocks.  $D$  denotes the domain pool, which specifies the set of the whole possible blocks of the size  $16 \times 16$  of image  $f$ , which makes up 58,081 blocks [21]. In the case of each block  $v$  from the range pool, fractal transformation (FT) is generated by searching all elements that exist within the domain pool  $D$  and exploring the most resembled block [26]. Let  $u$  signify a subsampled domain

block with a size similar to  $v$ . The similarity that exists between  $v$  and  $u$  is assessed by mean square error (MSE) which is shown as follows:

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M (u(i,j) - v(i,j))^2 \quad (1)$$

where  $M \times N$  is the image size.

FT allows the dihedral transformation of the domain blocks to be done. It is performed by rotating the domain blocks at angles  $270^\circ$ ,  $180^\circ$ ,  $90^\circ$ , and  $0^\circ$  (counter clockwise) and flipping the angle with respect to the point where  $y = x$ . In so doing, the eight orientations of the domain blocks could be generated. To achieve the most similar block from the domain pool, there exist 464,648 MSE computations for a particular block from the range pool. As a result, to encode completely the image by means of the basic algorithm of fractal compression (Full-search), 475,799,552 computations are totally required.

For a particular range block  $v$ , FT also allows the brightness  $q$  and the adjustment of the contrast  $p$  on the subsample domain block  $u$ . The similarity of  $u$  and  $v$  is determined by the quantity  $d = \|p \cdot u_k + q - v\|$ , where  $u_k$ ,  $1 \leq k \leq 8$  represents the eight transformations of  $u$ , in which  $q$  and  $p$  can be obtained as follows:

$$p = \frac{[N\langle u, v \rangle - \langle u, 1 \rangle \langle v, 1 \rangle]}{[N\langle u, u \rangle - \langle u, 1 \rangle^2]} \quad (2)$$

$$q = \frac{1}{N} [N\langle v, 1 \rangle - p\langle u, 1 \rangle] \quad (3)$$

where  $N = 64$ .

The fractal code of  $v$  is determined based on the location of the most similar domain block, the orientation  $k$ , the contrast  $p$ , and the brightness  $q$ . Practically, 16 bits are needed to represent the location of the domain block for a  $256 \times 256$  image. Given  $q$ ,  $p$ , and the orientation  $k$ , the bits values of 7, 5, and 3 bits are needed, respectively. The encoding process is completed when  $v$  completed execution across the 1,024 range blocks in  $R$ .

## 3 Imperialist competitive algorithm

Imperialist competitive algorithm (ICA) is an evolutionary algorithm for optimizing computation field. Atashpaz and Lucas [25] developed the ICA based on the evolution of human's socio-political societies. The ICA solutions are not strictly fixed at the local optima, which makes it applicable to problems requiring near-optimal solution. ICA was shown in [22] to be effective, especially in cases where a space of a problem exists among many local optima. This characteristic of the ICA makes it appropriate for FIC. Like other evolutionary algorithms, ICA can be defined by populating similar things into a group,

which can be referred to as ‘country’. Two groups of countries called the imperialists and colonies are determined from the population group. At the initial stage of this algorithm, some countries are generated and their powers are measured based on the cost function. Then, imperialists are selected among the most powerful countries, and others are considered as colonies of the imperialists. To assign colonies to the imperialists proportionally, the normalized cost of each is used. One of most significant issues in ICA is the assimilation policy where the countries move toward their imperialist (minimum optima). The assimilation requires that the colonies move individually on the line connecting them to their assigned imperialist [27].

Another factor that plays an important role in ICA is the imperialistic competition during which the empires with less power will lose their colonies, and by so doing, the colonies will be owned by other imperialists. For calculating the total power of an imperialist, two powers should be considered: its own power and that of its colonies. In each of the iteration, some colonies that are of less power are selected based on the revolution rate and replaced with new ones, randomly. Similar to other evolutionary algorithms, several conditions can lead to termination, which are when no more iteration occurs or there is no improvement to the final results.

#### 4 Proposed ICA-based fractal image compression

In this section, the strategy proposed in the current study is explained in detail. The strategy has two phases: first, partitioning all blocks of the image which is performed based on the edge property which is specified by the coefficients of DCT; and second, encoding each range block via ICA in which an appropriate mechanism for finding the best candidate empire is employed to decrease MSE computation.

##### 4.1 Phase 1: image partitioning based on DCT coefficient

The classification algorithm mainly aims at categorizing the whole image blocks (i.e., range and domain blocks) into three discrete classes. From the spatial domain, a block of image could be transformed to the frequency domain by means of DCT transformation. Within the frequency domain, the DCT coefficient that is situated in the upper left of the image block signifies the image block’s low-frequency information and its rough contour, whereas the DCT coefficient that is situated in the lower right of the image block signifies the image block’s high-frequency information and its fine texture. Therefore, we can explore the class of the image block by considering its lower-higher frequency DCT coefficients.

Therefore, for executing the classification, we used  $F_{(1,0)}$  which indicates the lowest vertical coefficient

and  $F_{(0,1)}$  which represents the lowest horizontal coefficient. The DCT function is presented as follows:

$$F_{(m,n)} = \frac{2}{N} C_m \cdot C_n \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} f(i,j) \times \cos\left(\frac{(2i+1)m\pi}{2N}\right) \cos\left(\frac{(2j+1)n\pi}{2N}\right) \quad (4)$$

where  $m,n = 0,1,\dots,N-1$ , and  $C_k = \begin{cases} 1/\sqrt{2} & \text{if } k = 1 \\ 1 & \text{otherwise} \end{cases}$ .

If a block is smooth, it means that it has a small  $F_{(1,0)}$  and  $F_{(0,1)}$ , so its block does not have a clear texture. Therefore, this block could be categorized as a smooth type and represented by S type. A far distance between  $F_{(1,0)}$  and  $F_{(0,1)}$  indicates the edge in the block. If  $F_{(1,0)}$  is large and  $F_{(0,1)}$  is small, it indicates a vertical edge. On the contrary, if  $F_{(1,0)}$  is small and  $F_{(0,1)}$  is large, it indicates a horizontal edge. In both conditions, we classify this image block as a horizontal/vertical edge that is represented by HV type. In another condition, if  $F_{(1,0)}$  is large and also  $F_{(0,1)}$  is large, it indicates that this block is either a subdiagonal or a diagonal edge. In this situation, this block is classified as DS type.

For determining the types of image blocks, first, the DCT transformation is executed for finding their  $F_{(0,1)}$  and  $F_{(1,0)}$  values that, in turn, can indicate the class of the blocks. For each image block, a DCT absolute coefficient pair ( $|F_{(1,0)}|$ ,  $|F_{(0,1)}|$ ) is constituted based on absolute values of its  $F_{(1,0)}$  and  $F_{(0,1)}$ . After that, we should normalize these absolute coefficient pairs; thereafter, they are known as ( $|F_{(1,0)}|_N$ ,  $|F_{(0,1)}|_N$ ). The obtained values are labeled on a Cartesian coordinate system in which  $|F_{(1,0)}|_N$  and  $|F_{(0,1)}|_N$  are  $x$  and  $y$  coordinates, respectively. Figure 1 shows the coefficient pair before normalization. As we can see in Figure 1a, all points are scattered throughout the Cartesian coordinate system, while in Figure 1b, the whole points are located on the first quadrant. It was performed to facilitate the classification of the points. In Figure 2, normalized coefficient pairs that are placed upon the first quadrant’s unit circle are presented.

For all points, the distance and angle are calculated from and to the first point of the Cartesian coordinate system.  $T_D$  and  $T_A$  are two thresholds that are used for classification;  $T_D$  is the distance threshold, and  $T_A$  is the angle threshold. Therefore, according to Figure 2, if an image block is located on the unit between  $T_A$  and  $90 - T_A$  and its distance value is more than  $T_D$ , it is classified as DS type. All blocks whose distance value is less than  $T_D$  are classified as S type. The remaining blocks are classified as HV class.

The algorithm of classification is shown as follows:

**Input:** Image blocks.

**Output:** Class type of image blocks.

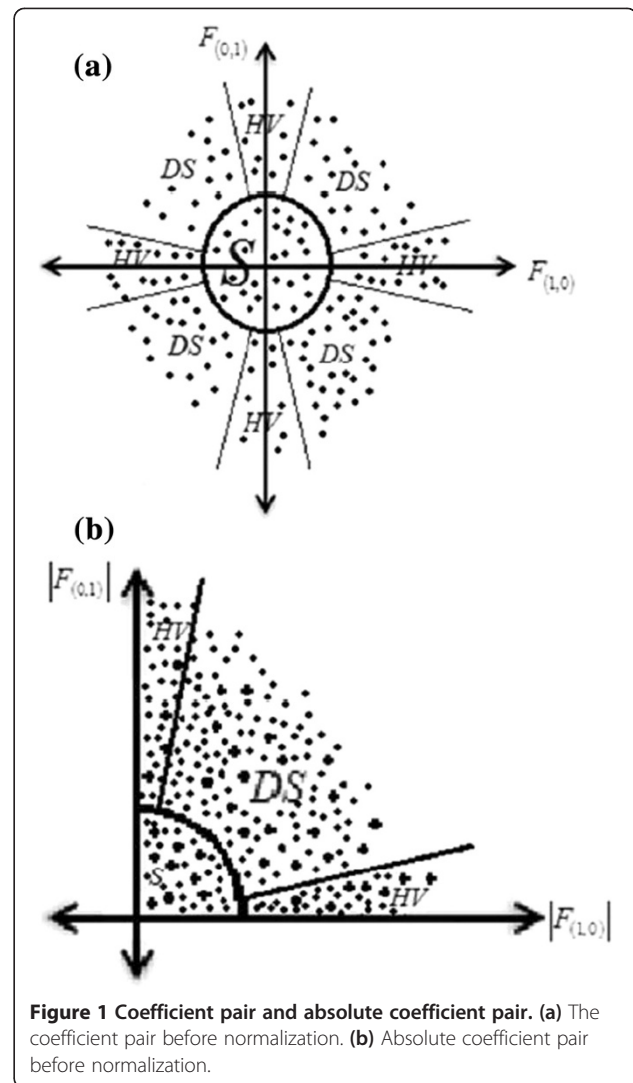
**Begin:**

01. Set NIB = number of image blocks.
02. **For**  $i=1$  **To** NIB **Do**
03.     Input an image block.
04.     FDCT =  $(F_{(1,0)}, F_{(0,1)})$
05.     AFDCT =  $(|F_{(1,0)}|, |F_{(0,1)}|)$
06.     **End For.**
07.     Normalize the AFDCT pairs.
08.     Put those in Cartesian coordinate system.
09.     **For**  $i=1$  **To** NIB **Do**
10.         Select  $P_i$
11.          $D = \text{sqrt}((|F_{(1,0)}|)^2 + (|F_{(0,1)}|)^2)$
12.          $A = (180/\pi)\text{arctg}(|F_{(1,0)}|/|F_{(0,1)}|)$
13.         **If**  $D < T_D$  **Then**
14.              $P_i$  belongs to  $S$  class.
15.         **Else**
16.             **If**  $(T_A \leq A \leq 90 - T_A)$  **Then**
17.                  $P_i$  belongs to DS class.
18.             **Else**
19.                  $P_i$  belongs to HV class.
20.             **End If.**
21.         **End If.**
22.     **End For.**

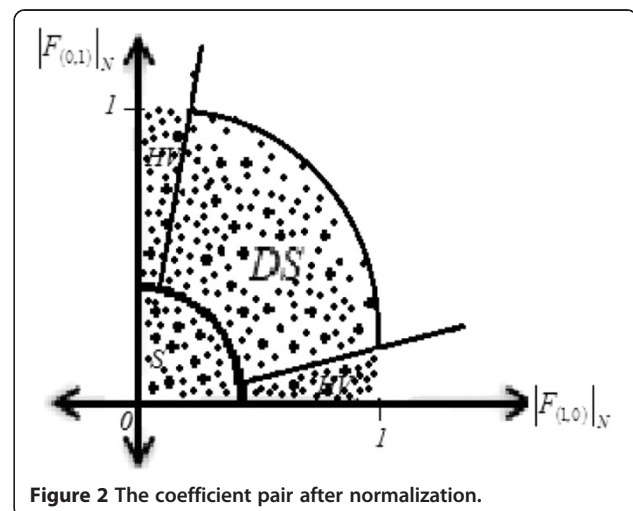
**End Algorithm.**

#### 4.2 Phase 2: DCT imperialist competitive algorithm for FIC

As it can be observed in the ICA algorithm, the values of power of countries (i.e., MSE) should be calculated for all countries existing in the algorithm. Based on the edge property of the blocks, the domain blocks will be considered as proper candidates if and only if they are in a class with the same label as that of the range block. To avoid the useless computations, countries could be divided into two groups: developed countries and undeveloped countries. A suitable country is one which is able to be matched with the range block. Countries of a certain class are more likely to be matched with the range block of the same class. These are developed countries



**Figure 1** Coefficient pair and absolute coefficient pair. (a) The coefficient pair before normalization. (b) Absolute coefficient pair before normalization.



**Figure 2** The coefficient pair after normalization.

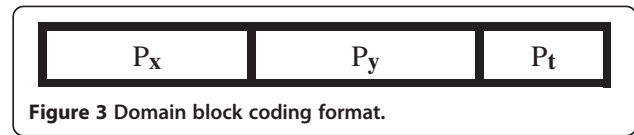
that have also the chance to be selected as empire. On the other hand, those countries whose class is not matched with the range block are known as undeveloped countries which, consequently, have less chance for being an empire. It is hoped that these countries will be developed and get a better position in the future decades. The power will be calculated merely for the developed countries. For the undeveloped countries,  $-Inf$  is assumed as their powers, which reduce MSE computation size. Based on the power of countries, the algorithm chooses NE countries as empire. Pseudo code of this proposed algorithm is shown below:

```

Input: Image.
Output: FIC codes.

Begin:
01 Set RN = number of range blocks, T = number of decades.
02 Set CN = number of all counties, NE = number of empires.
03 Set other parameters of ICA.
04 Calculate DCT and classify all blocks (range and domain).
05 For  $i=1$  To RN Do
06   Select  $R_i$ 
07   Make initial countries randomly.
08   Initial developing countries and undeveloped countries.
09   For  $k=1$  To CN Do
10     If (country is among the developing countries) Then
11       Calculate its power.
12     Else
13       Power of this country is  $-inf$ .
14     End if.
15   End for.
16   If (member of developed countries < NE) Then
17     Select from undeveloped countries randomly.
18     Calculate their power.
19     Add them to the set of the developed countries.
20   End if.
21   Rank the developed countries based on their power.
22   Decade = 1.
23   While (there is not just one empire and Decade < T) Do
24     Select best NE countries from developed countries.
25     Set them as empires.
26     Distribute remaining members of developed countries among empires based on their power.
27     Distribute undeveloped countries among empires based on their power.
28     For  $j = 1$  To NE Do
29       Move the colonies toward their relevant imperialist.
30       Form developed and undeveloped countries.
31       If (country is among the developed countries) Then
32         Calculate its power.
33       Else
34         Consider power of this country as  $-inf$ .
35       End if.
36       Exchange the position of the colony and empire.
37       If (revolution occur) Then
38         Select a country C
39         NC = create a new country randomly.
40         If (C is in developed countries and NC is in undeveloped countries) Then
41           Do not change position of C.
42         Else
43           C = NC.
44         Calculate its power
45         End if.
46       Compute the total power of imperialist.
47     End For.
48     Competition.
49     Eliminate the powerless empires.
50     Decade = Decade + 1.
51   End While.
52   Save the best empire parameters for range block  $R_i$  .
53 End For.

End Algorithm.
    
```



The stages of this pseudo code are explained as follows:

> Country formation

The stages of formation are a key part of DCTICA of FIC; in order to create the fractal encoding for the country using the partitioned iterated function system (PIFS), dihedral transformation  $p_t$  and the location  $(p_x, p_y)$  of the domain block for each country are considered. As demonstrated in Figure 3, the country has three parts that are allocated  $p_x, p_y$ , and  $p_t$ , where  $p_x$  represents the domain blocks in the horizontal direction,  $p_y$  its vertical position, and  $p_t$  the transformation at rotation  $0^\circ, 90^\circ, 180^\circ, 270^\circ$ , horizontal, vertical flip, relative to  $135^\circ$ , and flip relative to  $45^\circ$ .

> Calculate the country power

For calculating the power of each country within the population, the similarity characteristic is measured using MSE. The countries with less MSE are more similar to the candidate range block, and consequently, they will be more powerful.

> Initializing the empires and their colonies

More powerful countries are selected to be emperors; other counties are designated as colonies, which are dominated by one of the emperors. The colonies are distributed between the empires but on the basis of the empires' power; empires with more power will obtain more colonies. Based on the power of countries, the algorithm chooses NE countries as empire. As a result, three conditions may arise: (1) If NE is greater than the total number of developed countries, all these countries will be assigned as empire and the remaining empires will be selected randomly among undeveloped countries. (2) If NE is smaller than the number of members of the developed countries, NE members are selected as the empire

**Table 1** The value of  $T_A$  and  $T_D$

	Lena	Pepper	F16	Baboon
$T_A$	17.65	17.75	17.75	17.75
$T_D$	0.0559	0.0535	0.0469	0.0782

**Table 2 The number of elements of classes for different images**

Image	Range class			Domain class		
	S	HV	DS	S	HV	DS
Lena	310	329	385	10,495	21,149	26,437
Pepper	318	346	360	8,808	22,887	26,386
F16	464	263	297	20,569	17,730	19,782
Baboon	179	384	461	8,618	22,735	26,728

and the remaining will be distributed as colony among the empire. (3) If NE is equal to the developed countries, all will be selected as empire. The undeveloped countries will be distributed ultimately among the emperors according to the empires' power.

- Moving the colonies toward their relevant imperialist

Here, the integration policy that each country requires in order to move toward their local minimum optima is determined. As the emperor is more powerful than its colonies, optimal solutions are located more probably around them. Therefore, each colony should move toward its corresponding emperor. In essence, the colonies have the chances of getting a new position that may be better than their last position.

- The colony position exchange

In new positions, developed countries and undeveloped countries are formed. The power of developed countries is calculated merely. For the undeveloped countries,  $-Inf$  is assumed as their powers. If a colony possesses a higher power in comparison with its imperialist, that colony can become the new imperialist, and this can be termed exchange.

- Revolution

In each imperialist, according to the revolution rate, a country can be selected and is then moved to a new

position. This stage is performed randomly. In the proposed algorithm, when a revolution takes place in a country, it is hoped that this country would be directed to a better position. As we know, better situation for the country is that it could join the developed countries group. Therefore, if the revolution takes place in a developed country, and after revolution, it is positioned as an undeveloped one, and then we can ignore the revolution and consider its former position as the current one. On the other conditions, the new position obtained by the countries after revolution would be recorded for them.

- Computing the total cost

Here, for computing the total cost of each imperialist, first, we calculate the mean power of the developed countries belonging to that empire, and then we add a proportion of this value to the power of the empire itself. Equation 5 presents this operation.  $n$  stands for the number of developed countries of this imperialist, and  $\xi$  is a number considered to be between 0 and 1.

$$\text{Total power} = \text{Power of emperor} + \xi \left[ \left( \sum_{i=1}^n PC_i \right) / n \right] \quad (5)$$

- Competition

In this step, the weak colonies from the weak empires are picked and then added with high power imperialist.

- Eliminating the powerless empires

The weakest empires are those with no colony. If an empire is the weakest, its imperialism will be eliminated and replaced by a country that we consider as a colony with powerful empire.

**Table 3 MSE and PSNR for ICA, DCTICA, Vences-GA, Wu-GA1, and Full-search**

Method	Image name							
	Lena		Pepper		F16		Baboon	
	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR
Full-search	475,799,552	28.91	475,799,552	29.84	475,799,552	25.21	475,799,552	20.15
Vences-GA	1,148,135	25.62	1,148,159	26.30	1,148,169	22.47	1,148,148	19.25
Wu-GA1	2,313,986	27.15	2,376,360	27.91	2,357,251	23.91	2,314,056	19.68
ICA	1,625,065	27.39	1,621,543	28.26	1,601,334	24.27	1,583,434	19.76
DCTICA	806,667	27.28	742,224	28.08	790,945	24	746,372	19.68

**Table 4 MSE and PSNR for ICA, DCTICA, Vences-GA, Wu-GA1, and Full-search**

Method	Image name							
	Lena		Pepper		F16		Baboon	
	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR
Full-search	475,799,552	28.91	475,799,552	29.84	475,799,552	25.21	475,799,552	20.15
Vences-GA	1,436,415	25.89	1,436,450	26.45	1,436,439	22.70	1,436,451	19.28
Wu-GA1	2,977,756	27.30	3,073,367	28.06	2,955,752	23.99	2,907,500	19.71
ICA	2,043,976	27.70	2,050,406	28.42	2,028,660	24.35	2,028,030	19.80
DCTICA	1,026,341	27.38	943,734	28.19	1,008,198	24.04	947,014	19.74

**Table 5 MSE and PSNR for ICA, DCTICA, Traditional GA, Wu-GA2, and Full-search (GA without elitism)**

Method	Image name							
	Lena		Pepper		F16		Baboon	
	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR
Full-search	475,799,552	28.91	475,799,552	29.84	475,799,552	25.21	475,799,552	20.15
Traditional GA	5,120,000	26.23	5,120,000	27.10	5,120,000	23.32	5,120,000	19.47
Wu-GA2	2,831,391	26.25	2,361,127	26.65	2,653,875	23.01	2,501,905	19.34
ICA	2,589,860	27.81	2,599,350	28.55	2,568,558	24.41	2,546,793	19.88
DCTICA	1,288,816	27.52	1,196,451	28.38	1,275,981	24.35	1,196,809	19.78

> Terminating condition

Steps 23 to 51 would be repeated until Decade = 0 or just one imperialist would remain in the cycles. This empire is considered as the best solution for the supposed range block. For each range block, steps 6 to 53 are done.

### 5 Experimental results

Some tests are conducted on the proposed algorithm, and its performance has been compared to that of the Vences-GA [12], Wu-GA1 [20], Wu-GA2 [21], ICAFIC, and basic algorithm of fractal compression (Full-search) in terms of the number of MSE computation and peak signal-to-noise ratio (PSNR). The application of the proposed approach is checked on standard images F16, Pepper, Baboon, and Lena with the size of  $256 \times 256$  in gray-scaled mode. The size of range blocks and domain blocks is fixed at  $8 \times 8$  and  $16 \times 16$ , respectively. The PSNR is applied for comparing the results.

$$\text{PSNR} = 10 \times \log(255^2/\text{MSE}) \quad (6)$$

Parameters in ICA are set by  $\gamma = .0.5$ ,  $\beta = 1$ , and both the number of decades and the number of initial

imperialists are set to 10. For each chromosome, the crossover rate in GA is fixed to 0.6 and the probability of mutation to 0.01. The angle threshold for the Lena picture is set to 17.65, and for other pictures, it is set to 17.75. Table 1 presents the values of  $T_A$  and  $T_D$  for each picture. These values have been obtained via several experiments. Table 2 shows the number of elements existing in each class of range blocks and domain blocks. The results obtained from the experiments carried out on the proposed algorithm and GAs are demonstrated in Tables 3, 4, 5, and 6. In Table 3, the parameter of ICA for number of initial countries and parameter of GA for the number of population are set to 160, whereas in Table 4, they are set to 200, and in Tables 5 and 6, they are set to 250. The results presented in Table 6 obtained from GA are based on elitism.

For more evaluation of the algorithm performance, we set the number of initial countries by 300 for the image of Lena, and the results are presented in Table 7 and Figure 4.

According to Tables 3, 4, 5, 6, and 7, the proposed algorithm has been successful in improving the performance of FIC. As Tables 3, 4, 5, 6, and 7 clearly demonstrate, DCTICA has a better performance than the GAs, even those that are based on the elitism. Obviously, ICA could be suitably applied in FIC.

**Table 6 MSE and PSNR for ICA, DCTICA, Traditional GA, Wu-GA2, and Full-search (GA with elitism)**

Method	Image name							
	Lena		Pepper		F16		Baboon	
	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR	MSE computation	PSNR
Full-search	475,799,552	28.91	475,799,552	29.84	475,799,552	25.21	475,799,552	20.15
Traditional GA	2,688,000	27.46	2,688,000	28.29	2,688,000	24.11	2,688,000	19.72
Wu-GA2	1,463,472	27.03	1,240,140	27.90	1,329,282	23.86	1,319,639	19.52
ICA	2,589,860	27.81	2,599,350	28.55	2,568,558	24.41	2,546,793	19.88
DCTICA	1,288,816	27.52	1,196,451	28.38	1,275,981	24.35	1,196,809	19.78

Although in terms of the PSNR, the Full-search algorithm is better than the proposed algorithm; the proposed algorithm has a lower computational complexity, so it could be more advantageous than the other one. As an example, for Lena image, the Full-search method performs 475,799,552 MSE calculations, while the proposed method, with 200 initial countries and 10 initial empires, performs 1,026,341 MSE computations; it means about 463 times less than the Full-search algorithm. The method proposed in this research obtains an acceptable result with much less computational complexity. The PSNR per initial imperialist for Lena image shows in Figure 4 that the number of initial countries is set by 300.

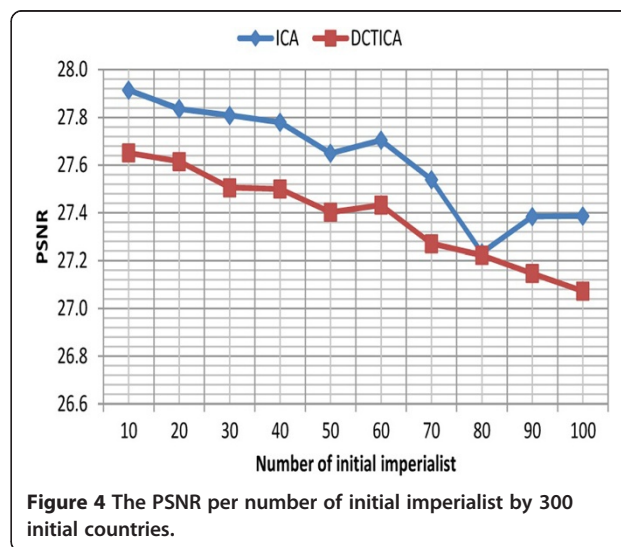
**6 Conclusion**

The main problem of FIC is the high number of MSE computations that has classified it in the NP-hard problems. In this paper, for reducing the MSE computation,

a two-phase algorithm is proposed. In the first phase, all image blocks were divided into three classes according to the image blocks' edge property. In the second one, the ICA algorithm found the suitable domain blocks using the results obtained in the first phase. For accelerating the algorithm, countries were divided into two groups - developed and undeveloped countries - and the value of MSE is computed only for the developed ones. Finally, results of the experiments carried out on the standard images demonstrated an improvement on FIC. Moreover, these results revealed that our method performance was more successful than the Wu-GAs and Vences-GA methods in terms of the retrieved image quality and the amount of MSE computation. It is noticeable that the amount of MSE computation in the Baboon picture has been 502 times more than that of the basic FIC algorithm, while the PSNR value is just 0.41 less than that of the basic FIC algorithm.

**Table 7 MSE and PSNR for ICA, DCTICA with initial countries by 300**

Number of initial imperialists	ICA		DCTICA	
	MSE computation	PSNR	MSE computation	PSNR
10	3,135,987	27.91	1,561,822	27.65
20	3,023,818	27.84	1,494,468	27.62
30	2,911,301	27.81	1,449,099	27.51
40	2,804,962	27.78	1,383,162	27.50
50	2,706,549	27.65	1,332,978	27.40
60	2,602,378	27.70	1,258,932	27.43
70	2,479,782	27.54	1,155,912	27.27
80	2,372,867	27.23	1,068,899	27.22
90	2,261,582	27.38	1,006,371	27.15
100	2,136,186	27.39	979,334	27.07



**Figure 4 The PSNR per number of initial imperialist by 300 initial countries.**



### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-264.

### Author details

<sup>1</sup>Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Malaysia.

<sup>2</sup>Computer Science Department, College of Computer & Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia. <sup>3</sup>MIS Department, CBA, Salman Bin Abdulaziz University, Alkharij 11942, Kingdom of Saudi Arabia. <sup>4</sup>College of Computer and Information Sciences, Prince Sultan University, Riyadh 12435, Kingdom of Saudi Arabia.

Received: 8 October 2013 Accepted: 16 April 2014

Published: 12 July 2014

### References

1. A Kaarna, P Zemcik, H Kalviainen, Multispectral image compression, in *Proceedings of Fourteenth International Conference on Pattern Recognition* (Brisbane, Australia, 1998), pp. 16–20
2. A Rehman, T Saba, *Evaluation of artificial intelligent techniques to secure information in enterprises* (Artif. Intell. Rev, 2012). doi:10.1007/s10462-012-9372-9
3. A Rehman, T Saba, Neural network for document image preprocessing. *Artif. Intell. Rev.* **42**(2), 253–273 (2012). doi:10.1007/s10462-012-9337-z
4. M Barnsley, L Hurd, *Fractal Image Compression* (A K Peters, Massachusetts, 1993)
5. A Jacquin, Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Proces.* **1**, 18–30 (1992)
6. Y Fisher, *Fractal Image Compression: Theory and Application* (Springer, New York, 1995)
7. Y Fisher, *Fractal Image Encoding and Analysis* (Springer, New York, 1998)
8. T Truong, C Kung, J Jeng, M Hsieh, Fast fractal image compression using spatial correlation. *Chaos, Solitons Fractals* **22**(5), 1071–1076 (2004)
9. T Truong, J Jeng, A fast encoding algorithm for fractal image compression using the DCT inner product. *IEEE Trans. Image Process.* **9**(4), 529–535 (2000)
10. S Furoo, O Hasegawa, A fast no search fractal image coding method. *Signal Process. Image Commun.* **19**(5), 393–404 (2004)
11. X Yang, D Liang, An improved genetic algorithm of solving IFS code of fractal image, in *3rd International Conference on Signal Processing* (Beijing, China, 1996), pp. 14–18
12. L Vences, I Rudomin, Genetic algorithms for fractal image and image sequence compression. *Comptacion Visual* **1**(1), 35–44 (1997)
13. X Chen, G Zhu, Y Zhu, Fractal image coding method based on genetic, in *International Symposium on Multispectral Image Processing* (Beijing, China, 1998)
14. S Mitra, C Murthy, M Kundu, Technique for fractal image compression using genetic algorithm. *IEEE Trans. Image Process.* **7**(4), 586–593 (1998)
15. X Lu, Z Yu, The application of GA in fractal image compression, in *Proceedings of the 3rd World Congress on Intelligent Control and Automation* (Hefei, China, 2000)
16. A Gafour, K Faraoun, A Lehireche, Genetic fractal image compression, in *ACS/IEEE International Conference on Computer Systems and Applications* (Tunis, Tunisia, 2003)
17. F Mohamed, B Aoued, Speeding up fractal image compression by genetic algorithms. *Multidim. Syst. Sign. Process.* **16**(2), 217–236 (2005)
18. X Lifeng, L Zhang, A study of fractal image compression based on an improved genetic algorithm. *Int. J. Nonlinear Sci.* **3**(2), 116–124 (2007)
19. M-S Wu, W-C Teng, J-H Jeng, J-G Hsieh, Spatial correlation genetic algorithm for fractal image compression. *Chaos, Solitons Fractals* **28**(2), 497–510 (2006)
20. M-S Wu, J-H Jeng, J-G Hsieh, Schema genetic algorithm for fractal image compression. *Eng. Appl. Artif. Intel.* **20**(4), 531–538 (2007)
21. M-S Wu, Y-L Lin, Genetic algorithm with a hybrid select mechanism for fractal image compression. *Digital Signal Process.* **20**(4), 1150–1161 (2010)
22. F Gao, G Wei, A fast coding algorithm based on variance and irregular region, in *Third International Conference on Intelligent Control and Information Processing (ICICIP)* (Dalian, China, 2012)
23. M Jayamohan, K Revathy, Domain classification using B+trees in fractal image compression, in *Conference on Computing and Communication Systems (NCCCS)* (Durgapur, West Bengal, India, 2012)
24. J Wang, N Zheng, 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 (2013)
25. E Atashpaz, C Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in *IEEE Congress on Evolutionary Computation* (Singapore, 2007)
26. A Qorbani, N Ali, A Ali, N Saeed, Square function for population size in quantum evolutionary algorithm and its application in fractal image compression, in *Sixth International Conference on Bio-Inspired Computing Theories and Applications (BIC-TA)* (Penang, Malaysia, 2011), pp. 27–29
27. T Saba, A Rehman, Effects of artificially intelligent tools on pattern recognition. *Int. J. Mach. Learn. Cybern.* **4**(2), 155–162 (2012)

doi:10.1186/1687-6180-2014-112

**Cite this article as:** Nodehi et al.: Intelligent fuzzy approach for fast fractal image compression. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:112.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)