

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,000

Open access books available

125,000

International authors and editors

140M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Optimization of Fractal Image Compression

Rafik Menassel

Abstract

Fractal encoding is a promising method of image compression. It is built on the basis of the forms found in the image and the generation of repetitive blocks based on mathematical translations. The technique seems to be moved theoretically and practically, but it requires enormous programming time due to the excessive resources required when compressing large volumes of data. On the other hand, metaheuristics represent all of the methods used to solve difficult optimization problems with less consumption of resources. They are marked by their rapid convergence and their lessening in research difficulties. In this chapter, we have tried to apply a new experience around the performance of organic metaheuristics inspired by nature, which are, respectively, the wolf pack algorithm (WPA) and the bat-inspired algorithm (BIA), as bioinspired techniques to optimize the fractal image compression (FIC). Experiments show the enhancement of diverse characteristics (coding time, compression rate (CR), peak signal-to-noise ratio (PSNR), and mean square error (MSE)). In addition, an assessment of the proposed approaches via many other approaches highlights this improvement.

Keywords: fractal image compression (FIC), metaheuristics, wolf pack algorithm (WPA), bat-inspired algorithm (BIA), image quality

1. Introduction

Nowadays, a significant size of information is managed and transmitted, and mainly, images have involved prodigious status, particularly in recognition field. Thus, it is important to decrease the size of the data via compression algorithms which can allow their storage and their transmission while using limited resources. Compression is employed to overcome this problem and keep more files. Mainly multimedia files need more storage space than other types of files. Images represent the largest part of the most used multimedia files in almost all fields. Unlike other types of files, a huge amount of image data requires more resources for storage and transmission on computer networks, and compression is therefore presented as an inevitable tool with the aim of more maneuverability of this data. Today, several compression formats exist while presenting their limits (degradations, size, duration, etc.) on somewhat particular images (text images and background areas).

To overcome this difficulty, scientists are constantly developing new techniques to compress images in order to find a perfectionist compression method that can largely conserve storage space and preserve the quality of the source file.

The compression methods that exist tend to introduce the theory of fractals, which appears to be a strong instrument for boosting image quality and reducing

the resources required. Nowadays, the image files intricated in several specific programs are characterized by massive data, pixel correlation, and similarity. Under such conditions, the old compression methods appear to be unsuitable for this mission because of their need for significant encoding time. On the other hand, the recently developed fractal image compression techniques offer better compression qualities [1–3]. These methods are built on the principle that shapes (fractals) can better represent usual scenes than traditional geometric forms. A number of fractal image compression techniques are presented in a lot of works; they include an image coding agreeing to more security and less degradation (Jeng et al. [4], Li [5], and Han [6]). These methods have revealed an important information, reduced statistical characteristics of chaotic patterns, and weakness in statistical cryptanalysis. In addition, metaheuristics are likewise employed to optimize compression, like genetic algorithms [7], ant colonies [8], and optimization by particle swarms [9]. These methods have ability to create a partition constructed on a region which improves the compression ratio and maintains better the decompressed image quality.

In this chapter, we try for the first time to apply natural metaheuristics on fractal compression, by suggesting new methods which associate, both the bat-inspired algorithm (BIA) and the wolf pack algorithm (WPA) with fractal image compression (FIC) to speed up encoding and optimize both file size and image quality. The main objective of using these algorithms is its property of search for global solution and its capacity to generate very satisfactory results quickly and for less means compared to similar techniques. The algorithms also use fewer parameters and without initial approximation of the unidentified parameters. This document is organized as follows; Section 1 presents an introduction in the study context. Next, a summary of the fractal image compression is detailed in Section 2. Section 3 summarizes the natural metaheuristics and presents those used in this work, namely, the WPA and the BIA. In Section 4, we give a review of some related works. After that, we present our proposed methods in Sections 5 and 6. Experiments are explained in Section 7. Finally, the conclusion appears in Section 8.

2. Fractal encoding

Fractal compression is a new method of irreversible image compression [10]. It was adapted by *Hutchinson* [11] and *Barnsley and Demko* [12]. It searches for self-similarities among the diverse image blocks [13] and only keeps the parameters of the contractual transformation in place of the pixels of the image. Like this, we can build an estimate as close as possible to the source image by detecting the redundancy of forms at several scales and try to eliminate these redundancies in the original image so that the result is precise enough to be accepted.

The FIC is founded on an iterated function system f , a limited group of contractions defined on a metrical space R_n by the relation:

$$f_i : R_n \rightarrow R_n; i < N \quad (1)$$

This contraction can be in several shapes depending on technical constraints. It can be done several points of the original image and carry them nearer to the compressed one. This reflection is named “affine transformation”; then, each sub-block of the original image will be submitted either a rotation at an angle, a scale, or a translation (transformed using eight isometries) according to Equation 2:

$$w(x) = T(x) + b \quad (2)$$

where T is a linear transformation, $R_n \rightarrow R_n$ is a vector, and $b \in R_n$ is a vector. Practically, the general principle of fractal compression is to try to find the finest matching domain blocks for each range block in order to minimize the distance metric. The code that follows illustrates this idea well:

- Enter the original image
- Create a partitioning of Range Blocks R
- Create a partitioning of Domain figures
- **For** all figures Destination Make
 - **For** all figures Sources Make
- **For** all the defined transformations Make
 - Apply the transformation to the Range Blocks
 - Adjust the average of the pixel colors
 - Apply the reduction from the Range Blocks to the Domain Blocks
 - Calculate the error between the result and the Domain Blocks
 - **If** the error is minimal for the destination figure **Then**
- Save the modifications made
 - **End if**
- **End For**
- Write the saved values in the output file
 - **End For**
- **End For**

This process is realized following the relation:

$$B_i = v(D_i) \quad (3)$$

where $v()$ is the function of the contraction which aims to modify D_i . Thereafter, the nearest block B_i is sought for all R_i blocks by calculating the error between B_i and R_i , and we can use, for example, the Hausdorff distance defined by Equation 4:

$$H(B_i, R_i) = \max(d(B_i, R_i), d(R_i, B_i)) \quad (4)$$

where $d(B, R) = \max_{b \in A} \min_{r \in B} \|b-r\|$.

Or using the Euclidean distance described by the equation:

$$d^2(R, B) = \sum_{i=1}^n (r_i, b_i)^2 \quad (5)$$

where n represents the pixel's number in R_i and B_i blocks. d should be as minimum as possible for blocks that look alike.

Fractal decompression consists of the reconstruction of R_i from the blocks B_i which appear identical the most by practicing the contraction used in compression.

In recent years, several studies have concentrated on the development of accelerated decryption procedures [14–16] with the aim of preserving image quality. Their principle is to choose a random image as the original one and execute an affine transformation like defined in Equations (6) and (7), founded on the fractal ciphers obtained by itself. This act is repeated recursively till the reconstructed image were satisfactory:

$$R_i = S_i \cdot D + O_i \cdot I \quad (6)$$

$$S = U R_i \quad (7)$$

where I is a contractive or isometric spatial transformation, D is a domain block, R is a range block, and S is the reconstructed image.

In fact, FIC is becoming among the most promising methods for image compression for its significant compression ratio (CR) and preservation of quality. Its beginnings date from the 1990s when Jacquin [17, 18] introduced the first method of image compression; its principle is partitioning the image into two tiling blocks: the range and domain blocks.

The domain blocks are double the size of the range blocks and overlap such that a new domain block starts at each pixel. The main idea of this compression is to find the nearest domain block in concordance with each range block, to determine the right contractual transformation, and to store all these parameters. This principle was exciting; however it remained limited to local applications because it consumes a lot of CPU time. Since then, researchers have constantly presented new techniques to reduce the compression time; Thomas and Deravi [19] link range blocks and brand them more adaptive with image content by using the region-growing method. Cardinal [20] presented an alike principle; it is founded on a geometrical partition of the grayscale image block feature space. The experimental evaluations with earlier published methods illustrate an important enhancement in encoding time with practically better quality. *He et al.* [21] have used the normalized block with the aim to evade the extreme search in corresponding block. Chong and Pi [22] proposed a new adaptive search method to decrease the calculation complexity of fractal encoding to discard a big number of unqualified domain blocks so as to speed up FIC.

Other studies have been presented on new aspects to improve the way of research like the encoding via the Fourier transform [23], special image features [24], and discrete cosine transform inner product [25]. The majority of existing methods rely on a corresponding error threshold to limit the search. Lately, Lin and Wu [26] have defined another way of search built on image block edge property, which proves suitable results. Furthermore, many research articles have been published over the past decade; they increased the quality of the compressed image through the use of metaheuristics without resorting to more resources in the coding process.

3. Natural heuristics

Heuristics refer to the set of techniques that can solve several problems by maximizing gains and decreasing the resource's consumption; however, the optimal

solution cannot be certain if, in the investigation space, there is an intersection between the local and global solution [27]. Natural heuristics represent a large family of heuristics which are inspired from communal conduct of animals existing in societies like assemblages of birds, ant colonies, or grouping of fish. They are founded on the principle of populations of entities who cooperate and develop rendering to reciprocal precepts. These techniques allow the invention of procedures which can resolve hard problems by dividing control. These approaches form a famous prototype which is effectively employed as a prodigious tool for resolving difficult problems [28] with less resource consumption. Several researches [29, 30] illustrate that these systems have an effective potential to manage various situations and could be adapted to bring solutions to diversified optimization problems.

3.1 The wolf pack algorithm (WPA)

The wolf pack algorithm (WPA) [31] belongs to the family of bioinspired techniques which can be used to estimate resolutions for numerous optimization problems. WPA is a metaheuristic built on the population invoked by the social hunting behavior of wolves. It basically involves hunting wolves, tracking down prey, and capturing it under the orders of a leader wolf. The wolf pack includes the strongest and most intelligent wolf chef. He is responsible for controlling the pack. Its decisions are always based on the surrounding environment: prey, pack wolves, and other hunters. The pack is divided into two families of wolves: scoot and ferocious.

The scoot wolves move autonomously in the milieu and adjust its way according to the concentration of the odor of the prey. When a prey is found, scoot wolves cry and transmit info by sound to the leading wolf who guesses the distance to reach this prey; it calls the furious wolves and quickly displaces towards the cry. The prey is then caught and is shared conferring to the nature of each wolf: from the sturdiest to the feeblest. Subsequently, feeble wolves could die from absence of nutrition. In this manner the pack ensures a certain dynamic and robustness at all times.

WPA is performed as follows:

In a search environment \mathbb{R}^n , any wolf i denotes an elementary solution to the problem, at a location x_i .

Initially, wolves are distributed chaotically in the environment.

At any instant t , the wolf i passes from the location x_i^t to the location x_i^{t+1} . The choice of the following location is updated by rendering the following equation:

$$x_i^{(t+1)} = x_i^t + \lambda |x_g^t - x_i^t| \quad (8)$$

where λ represents a vector randomly distributed in the interval $[-1,1]$ and x_g designates the location of the chief wolf. After a static sum of repetitions, which corresponds to a research stage, the wolf of the finest result gets converted to a leader one; feeble wolves (bad solutions) will be wiped out and substituted with a novel group of wolves in an arbitrary manner.

3.2 The bat-inspired algorithm

The bat-inspired algorithm (BIA) [32] belongs to the family of metaheuristics inspired by nature, introduced by Yang and founded on the echolocation comportment of bats. Bats have a system identical to radar except that radars use electromagnetic waves while bats use ultrasonic waves (of frequency inaudible to humans). Bats move and hunt with high-performance sonar. By another way, bats are distinguished by an extraordinary steering mechanism allowing them to

differentiate between an obstacle and a prey, which allows them to hunt even all. The impulses produced by bats can be linked almost to the hunting plans of bats. Often, the pulses are between 25 and 150 kHz at a static frequency; they only persist for 8–10 ms. Bats generate between 10 and 20 ultrasonic sound eruptions every second, each of them stays between 5 and 20 ms. However, when bats seek their prey and feel so close, they can increase the rate of emission of sound eruptions up to 200/s. This proves the extraordinary ability of bats to process signals. Assuming that the speed of sound in air is $v = 340$ m/s, the wavelength λ of the ultrasonic sound therefore manifests with a continuous frequency f :

$$\lambda = v/f \quad (9)$$

The wavelengths vary between 2 and 14 mm and are equivalent to the size of the bat's prey, for a representative frequency between 25 and 150 kHz. The pulses produced by bats can spread an imposing sound intensity of 110 dB, but quite favorably, these pulses remain in the ultrasonic domain. The intensity of the pulse can take different stages, such as very strong when bats are chasing and weak at a quiet sound when they mark their prey. These short pulsations usually have a wandering range of a few meters which depends on the frequency.

In reality, bats combine all of their senses to effectively detect prey and navigate more easily. Here, we are only interested in echolocation and the behaviors that accompany it. To create new optimization techniques, the echolocation of bats can be transformed into an optimized objective function.

In a search space R_i , the bats fly randomly using the speed V_i in location (solution) X_i using velocity V_i . They produce pulses at a static wavelength λ with a variable frequency f and an intensity A (differs from a big positive A_0 to a smallest constant value A_{\min}) to hunt for prey. When the bats choose the finest results, they choose a local result from the best selected ones.

4. Introducing bioinspired approaches in the FIC

In 2005, Dervis Karaboga proposed a new iterative optimization method based on artificial bee colonies (ABC). This technique is based on three different classes of bees, (a) bee used, (b) spectator bee, and (c) scout bee. The spectator bees waiting in the store obtain data concerning the sources of nectar revealed earlier from the employees. Then they choose a usable nutrition source built on the received information. Scout bees arbitrarily search for nutriment in the area for [33].

In 2006, Cristian Martinez presented an enhanced image compression using the ant colony technique. The basic idea is that ants always seek and find the shortest path from nest to food source using the pheromone. For fractal compression, the pheromone is positioned on the range block i and the domain block j . The pheromone matrix is rectangular (not symmetrical) where the lines designate range blocks (image blocks) and the columns indicate domain blocks (blocks to transform). Then, the ants build routes by choosing a block of domain j for each block of range i . the solution will be found on the basis of updating the pheromone and heuristic information [34]. The result proposes similar image quality to that obtained with a deterministic way while minimizing the calculation time by 34%.

In 2009, many of studies were focalized on FIC: Chakrapani and Soundara Rajan [35] have created a new fractal image compression founded on a genetic algorithm in the intention of optimizing the encoding time for an acceptable image quality. The results give improved performance over exhaustive search.

In the work of Xing-yuan, Fan-ping, and Shu-guo [36], a spatial correlation hybrid genetic algorithm that uses the features of the fractal and divided iterative function system is proposed. It consists of two stages. The first uses spatial correlation in the images for the range and the domain blocks in order to exploit the local optima. The second one is based on a genetic-simulated annealing algorithm (SAGA) to find the global optima if the local optima is not satisfied. In order to escape early convergence, the algorithm approves that the dyadic mutation operator takes place in place of the traditional operator.

In 2010, Chakrapani et al. have enhanced the fractal image compression using particle swarm optimization (PSO) technique [37]. PSO is used to speed up the search of the nearest finest match block for a definite block to be encoded. This method illustrates that the recovered image quality can be conserved when in comparison with the full-search FIC.

In 2016, Shaimaa S. Al-Bundi et al. [38] use an upgraded genetic algorithm to enhance the exploration space in the target image by good estimation to the global optimum in an only execution.

In 2017, Al-Saidi N.M.G et al. [39] optimize the fractal image compression by introducing the harmony search algorithm. This strategy searches for the best solution through singing a song; this proposed technique offers splendid performances in terms of image quality, reduced computation time, and storage space when compared to other methods.

In 2018, we [40] used the wolf pack algorithm to improve the FIC; the idea is to take the entire image for the search space where this space is divided into blocks; scooters wolves roam the environment to find other smaller and similar blocks. They examine the entire space and select the blocks with the best physical shape. By this method, the encoding time was considerably reduced, and we also obtained a better compression rate.

In 2019, we have [41] chosen to improve the FIC by using the bat-inspired algorithm. Our tow proposed methods are detailed and well explained in Sections 5 and 6, respectively.

5. Wolf pack algorithm to optimize fractal image compression

We assume an image of $m \times n$ pixels as exploration space, represented by an array P where each pixel is considered as a cell and on a byte (gray pixel). The resulting image C of $m/2 \times n/2$ pixels is reached by following the steps:

- Divide the entire image into tiny nonoverlapping r_i blocks of size $s \times s$ (with $s \ll m$). More simply, we will proceed with blocks of square size of $b \times b$; this partition called range block will be represented by $R_N = \{r_1, r_2, \dots, r_N\}$.
- For all the blocks r_i , the scooting wolves roam the space in order to find a d_i of size $2b \times 2b$ similar with r_i while respecting the parameters mentioned above. A fitness value $f(d_i)$ will be assigned for each block d_i according to Eq. (6). The block d_i is taken for prey.
- After the hunting wolves have inspected the entire space and for all r_i blocks, d_i blocks with the best physical form are selected. It will be mapped according to Eqs. (4) and (5).

If no improvement is made to the wolf chef solution, the process will be stopped after a fixed number of iterations.

The adapted WPA algorithm for FIC is showed as:

Algorithm 1. FIC-WPA

```

Initialization:
Generate  $r_i$ , ( $i = 1 \dots N$ )
For each  $r_i$ ,  $f(r_i) \leftarrow 0$ , ( $i = 1 \dots N$ )
  While not (stopping criteria)
     $it \leftarrow 0$ 
    While not (Iter-scoot < It)
      Pick random numbers:  $\lambda \in [-1,1]$ 
      For each  $r_i$  do
         $x_i \leftarrow x_i + \lambda|x_g - x_i|$ 
        If ( $f(bx_i) < f(g)$ )  $g \leftarrow i$ 
      Endif
    Endfor
     $r_i \leftarrow v(di)$ 
    Update It
  EndWhile
End For.

```

6. BAT-inspired algorithm introduced in FIC

The similar idea employed in the wolf pack algorithm and the BAT algorithm for FIC is completed through the succeeding phases:

1. The whole image is scouted randomly by bats with the use of loudness L and frequency F .
2. Each block is compared to all of its neighboring blocks by bats for its degree of homogeneity as a function of volume and frequency. If they meet a criterion ($\text{color_level_block} - \text{neighboring color level} \leq \text{frequency}$), we create a domain block of size $L * L$ whose value is only the mean of the domain block.
3. When the bats roam the whole picture. The iteration will be stopped.
4. After decomposing the image into domain blocks, the bats' position themselves at x_i , and the size of the blksz block will be saved in a sparse S .
5. By eliminating the solution with the smaller block, we try to find the best solution in this step.
6. In order to calculate the compression ratio, we will use Huffman encoding to store all information (locations, block sizes, and values).
7. Finally, and to reconstruct the image, we will use Huffman decoding to regenerate the image data of the compressed image.

Algorithm 2. Proposed FIC-BAT

```
Initialization: Generate bats (Number_bats = 1..N) //
Begin
Loudness L;
Frequency F;
While not (stopping criteria)
  For each bat
    If similarity (distance) = 1
      Create domain block;
      Store location in vectors I,J;
      Store block sizes in vector blkosz;
    Else
      Store location in vectors I,J;
      Store block sizes in vector blkosz;
    End-if
  End For
End-while
Search for best solutions;
Store the locations and block sizes in a sparse S;
End.
```

7. Experimentations

7.1 Fractal image compression with WPA

The resolution presents an aspect that should not be ignored when trying to test the efficiency of an approach. For our circumstance, we will take into consideration the three test images (Lena, Barbara, and cameraman) with different resolutions (16 * 16, 32 * 32, 64 * 64, 128 * 128, 256 * 256) so that we can see the impact of this factor on the quality quantities (compression ratio, compression time, EQM, PSNR) (Figures 1 and 2).

From the test images with a reconstructed resolution of 256 * 256, it can be seen that the quality of the images is acceptable to a very good degree. And to be more exact, we must refer to calculable measures.

The table below presents the results obtained by applying our approach to the above images with different resolutions:

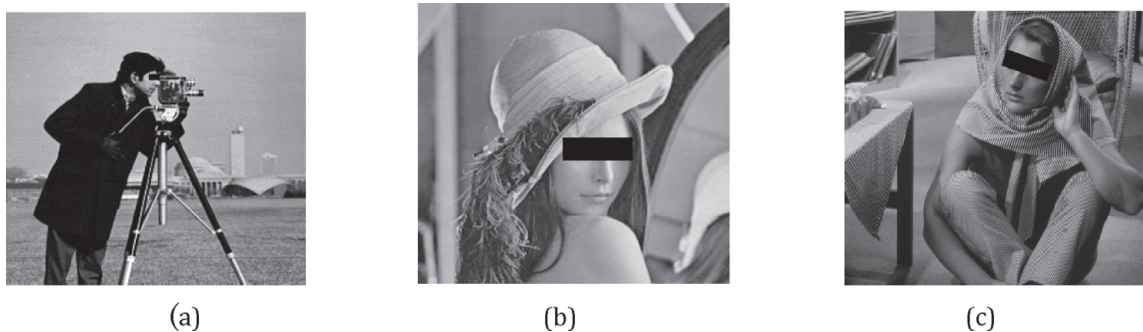


Figure 1.
Tested images (before compression). (a) Cameraman, (b) Lena, and (c) Barbara.

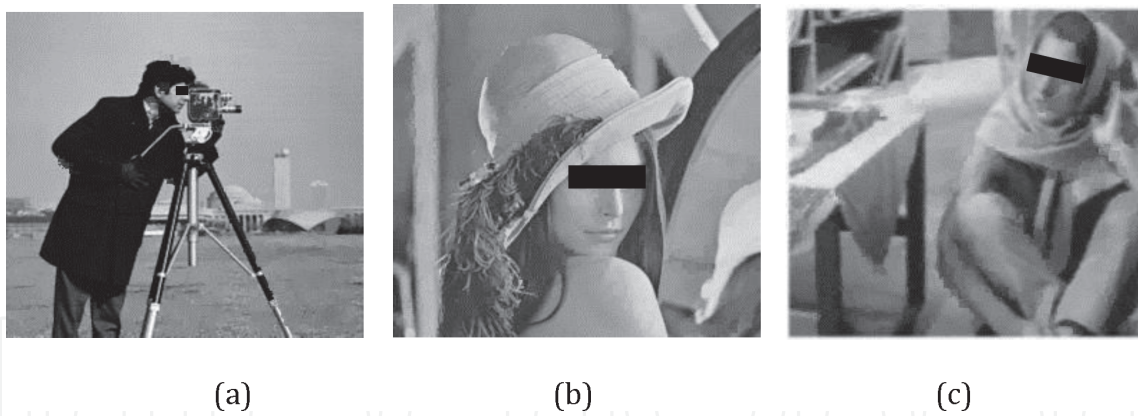


Figure 2. Decompressed tested images (after WPA compression). (a) Cameraman, (b) Lena, and (c) Barbara.

The results obtained (detailed in **Table 1**) illustrate firstly the agreement between image resolution and encoding time, which demonstrates that our proposed method is significantly responsive to image resolution; it is also clear that the image quality is in contrast linked to it (image quality degrades quickly as resolution increases). This is clearly shown by the amplitudes of the PSNR and the MSE.

The compression ratio remains relative to the resolution where we indicate that our method interferes in this measurement. Wolves appear when the quantity of blocks is very huge (higher resolution) by contributing a higher compression rate than that existing in lesser resolutions.

Decompression time stays optimized and is similar for almost all methods. This is due to the decompression process which is not complex compared to the compression process.

We will now focus on comparing our proposed method with some other techniques. The following table (**Table 2**) shows the notable alteration between our method and the others:

Tested image	Resolution	Compression time (s)	Decompression time (s)	Compression ratio	MSE	PSNR (dB)
Cameraman	16*16	0.184	0.818	1.098	2.617	37.999
	32*32	0.446	0.802	1.174	4.079	36.173
	64*64	2.451	0.816	1.420	6.853	31.509
	128*128	41.284	0.810	1.653	8.931	30.096
	256*256	774.438	0.897	1.741	9.152	31.605
Lena	16*16	0.185	0.832	1.095	7.730	33.616
	32*32	0.507	0.801	1.125	5.861	34.451
	64*64	2.290	0.791	1.287	8.401	33.550
	128*128	34.257	0.821	1.480	9.547	32.641
	256*256	668.810	0.850	1.596	9.282	33.305
Barbara	16*16	0.225	0.812	1.019	1.352	38.070
	32*32	0.607	0.763	1.055	4.064	36.067
	64*64	1.868	0.781	1.152	6.259	35.060
	128*128	22.713	0.793	1.310	11.093	32.348
	256*256	509.406	0.861	1.447	11.115	33.288

Table 1. Variation in image resolution.

Images de test	Methods	PSNR (dB)	Time (s)	Ratio (CR)
Lena 128*128	FIC-WPA	32.641	34.257	1.480
	Suman K. Mitra et al.'s works [7]	30.22	/	1.059
	Vishvas V. Kalunge et al.'s works [42]	/	67	/
Lena 256*256	FIC-WPA	33.305	668.810	1.596
	Y. Chakrapani et al.'s works [37]	26.22	2370	1.3
	Exhaustive search	32.69	8400	1.3
	DWSR [31]	25.8212	56.4247	1.56355
	PSO-RCQP [43]	27.089	6.453	1.6392
Cameraman 256*256	FIC-WPA	31.605	774.438	1.741
	PSO-RCQP [43]	26.686	268	1.8212
Barbara 128*128	FIC-WPA	32.348	22.713	1.310
	Vishvas V. Kalunge et al.'s works [42]	/	66	/
Barbara 256*256	FIC-WPA	33.288	509.406	1.447
	Y. Chakrapani et al works [37]	28.34	2500	1.2
	Exhaustive search	32.84	8400	1.2

Table 2.
 Comparison of the FIC-WPA with other techniques.

7.2 Using bat-inspired algorithm to enhance fractal image compression

Bat's number: In **Table 3**, we will examine the images of the cameraman and Lena in order to extract the adequate number of bats that will be used in our approach.

Loudness: **Table 4** presents some tests carried out on an additional stress which is loudness, with the objective of taking into consideration the good value for the test.

Frequency: The last test table (**Table 5**) is made to pick up the best frequencies that will be used in our algorithm.

Images	Number of bats	Compression time	Decompression time	Compression ratio	PSNR	MSE
Cameraman	2	0.488	0.705	1.385	31.608	10.088
	4	0.459	0.707	1.366	30.934	8.563
	8	0.452	0.729	1.372	31.216	9.417
	16	0.451	0.749	1.349	30.934	8.879
	32	0.509	0.843	1.355	29.827	10.045
	64	0.472	0.919	1.365	30.412	10.405
	128	0.478	0.749	1.348	31.895	9.663
	256	0.475	0.883	1.335	30.989	8.899
	512	0.457	0.750	1.392	29.997	9.870
Lena	2	0.518	0.739	1.303	30.629	14.440
	4	0.548	0.727	1.315	30.083	15.612
	8	0.505	0.716	1.311	30.071	14.929

Images	Number of bats	Compression time	Decompression time	Compression ratio	PSNR	MSE
	16	0.514	0.713	1.306	29.984	15.348
	32	0.563	0.784	1.299	30.452	15.037
	64	0.562	0.779	1.298	31.228	13.887
	128	0.512	0.715	1.306	30.669	15.603
	256	0.518	0.709	1.320	30.389	16.118
	512	0.520	0.702	1.321	29.856	14.710

Table 3.
Testing the number of bats.

As we can see from the previous tables, the best choices are given by 8 for the number of bats, 8 for the intensity, and 30 for the frequency, so the treatment parameters are as follows:

The number of iterations is in the interval of [10,100].

The number of bats is fixed at 8.

Loudness 8, Frequency 30.

We experienced the proposed approach on five images through diverse resolutions. **Table 6** illustrates certain quality measurement liable on the resolutions.

Image	Loudness	Compression time	Decompression time	Compression ratio	PSNR	MSE
Cameraman	2	0.453	0.720	1.267	33.788	6.531
	3	0.452	0.729	1.372	31.216	9.417
	4	0.458	0.721	1.376	33.022	7.807
	5	0.469	0.702	1.355	30.220	9.375
	6	0.438	0.727	1.359	30.119	9.027
	7	0.460	0.723	1.348	28.971	9.130
	8	0.442	0.727	1.279	34.284	5.600
	9	0.455	0.736	1.220	30.813	6.962
	10	0.468	0.757	1.205	33.859	5.012
	11	0.472	0.755	1.187	33.548	4.362
	Lena	2	0.530	0.735	1.248	31.781
3		0.505	0.716	1.311	30.071	14.929
4		0.537	0.726	1.324	29.230	16.853
5		0.586	0.751	1.299	30.536	14.382
6		0.519	0.713	1.244	30.812	13.766
7		0.516	0.715	1.223	31.290	12.190
8		0.503	0.756	1.228	31.032	11.917
9		0.495	0.740	1.204	32.213	9.056
10		0.492	0.761	1.194	31.873	8.372
11		0.488	0.737	1.192	32.050	9.695

Table 4.
Testing the values of loudness.

Image	Frequency	Compression time	Decompression Time	Compression ratio	PSNR	MSE
Cameraman	20	0.490	0.714	1.118	40.341	1.526
	30	0.458	0.698	1.209	33.576	4.4806
	40	0.442	0.727	1.279	34.284	5.600
	50	0.460	0.738	1.357	29.463	11.990
	60	0.435	0.706	1.422	26.887	15.954
	70	0.453	0.716	1.586	23.701	22.989
	80	0.467	0.702	1.722	23.769	33.415
	Lena	20	0.495	0.755	1.066	42.199
30		0.512	0.735	1.099	35.522	4.947
40		0.503	0.756	1.228	31.032	11.917
50		0.514	0.718	1.359	27.649	23.370
60		0.549	0.745	1.590	23.577	39.059
70		0.521	0.717	1.850	22.204	50.050
80		0.646	0.720	2.064	20.869	58.349

Table 5.
 Testing different values of frequency.

Test image	Resolution	Compression time (s)	Decompression time (s)	Compression ratio	MSE	PSNR (dB)
Blonde women	32*32	0.455	0.692	1.223	9.643	32.645
	64*64	2.581	0.698	1.430	13.100	31.678
	128*128	34.634	0.712	1.712	14.165	30.853
	256*256	811.291	0.763	1.741	13.150	32.065
Lena	32*32	0.522	0.704	1.109	5.651	34.695
	64*64	2.193	0.704	1.282	8.722	33.380
	128*128	33.376	0.720	1.486	9.799	32.909
	256*256	732.345	0.763	1.604	9.660	33.115
Cameraman	32*32	0.467	0.832	1.211	3.628	36.328
	64*64	2.608	0.719	1.440	7.743	30.258
	128*128	43.854	0.691	1.658	9.086	31.078
	256*256	732.011	0.977	1.678	9.626	31.095
Living room	32*32	0.558	0.687	1.219	13.475	31.210
	64*64	2.092	0.692	1.355	14.584	31.414
	128*128	26.347	0.710	1.487	14.584	31.322
	256*256	478.219	0.767	1.555	14.138	31.599
Mandrill	32*32	0.598	0.554	1.211	11.936	31.674
	64*64	2.376	0.889	1.422	18.377	30.555
	128*128	22.190	0.732	1.419	18.435	30.582
	256*256	334.636	0.795	1.368	16.404	30.737

Table 6.
 Testing the image resolutions.

Figure 3 shows the image of blonde women before and after proposed compression.

In **Figure 4**, we show a cameraman image before and after proposed compression,

Figures 5 and 6 describe separately Lena and living room images before and after applying the proposed compression.

Finally, we conclude our sequence of assessments with mandrill image before and after proposed compression, in **Figure 7**.

As we can observe, the images' quality is very suitable.

And to approve this effect, **Table 6** explores additional quality measure.

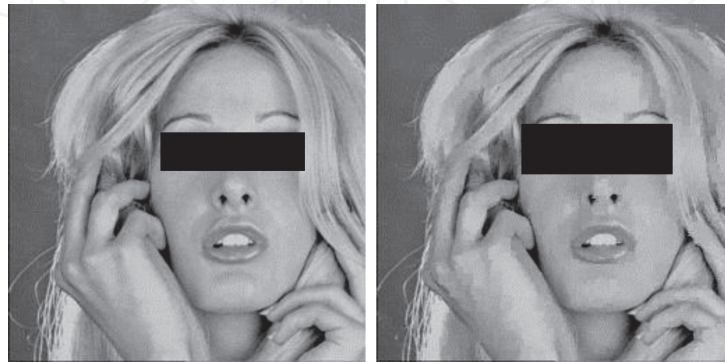


Figure 3.
An image of Compressed and decompressed blond women.



Figure 4.
An image of compressed and decompressed cameramen.



Figure 5.
An image of compressed and decompressed Lena.

The table shows that our approach is significantly sensitive to changing resolutions. It is also clear that the quality of the images is inversely linked to the resolution (as soon as the resolution increases, the quality of the images degrades) which is well proven by the MSE and PSNR measurements.



Figure 6.
 An image of compressed and decompressed living room.

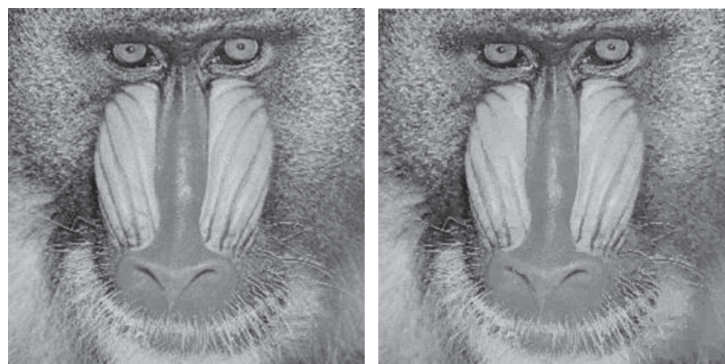


Figure 7.
 An image of compressed and decompressed Mandrill.

Test image	Methods	PSNR (dB)	Compression time (s)	Compression ratio
Lena 128*128	BIA	32.909	33.376	1.486
	Suman K. Mitra et al works [7]	30.22	/	1.059
	Vishvas V. Kalunge et al works [42]	/	67	/
Lena 256*256	BIA	33.115	732.345	1.604
	Y. Chakrapani et al.'s works [37]	26.22	2370	1.3
	Exhaustive search	32.69	8400	1.3
	DWSR [44]	25.8212	56.4247	1.56355
	PSO-RCQP [43]	27.089	6.453	1.6392
Cameraman 256*256	BIA	31.095	732.011	1.678
	PSO-RCQP [43]	26.686	268	1.8212
Barbara 128*128	BIA	32.176	21.478	1.312
	Vishvas V. Kalunge et al.'s works [42]	/	66	/

Table 7.
 Proposed approach versus other methods.

We note that our proposed method has a remarkable effect on compression ratio and remains relative to the resolution; the bats then show themselves, when the number of blocks is greater (more resolution) by offering a compression ratio greater than that present in the lower resolutions.

On the other hand, the decompression process does not include any complexity compared to the compression process which gives us a decompression time which remains optimized and similar for almost all resolutions.

Finally, and to be in the set of techniques which try to optimize fractal compression, we will draw up a comparison of our approach with certain existing methods. **Table 7** clearly shows the remarkable difference between our method and the others.

At first glance, our proposed method represents new work which has led to satisfactory results. It proportionally retains the quality offered after compression. Reduced time remains very satisfactory, and the compression rate is better than that offered by most of the methods below.

8. Conclusion

In this chapter, we explained the possibility of optimizing fractal image compression using bioinspired metaheuristics. We focused on the application and, for the first time, recent metaheuristics which are, respectively, the wolf pack algorithm and the bat-inspired algorithm in order to improve the performance of fractal image compression. The results demonstrate the efficiency of the algorithms considered.

Compared to other optimization metaheuristics, our approaches offer better results in many aspects, mainly the encoding and decoding time, the size, and the quality.

In addition, proposed approaches demonstrate the ability of these techniques to manage image compression.

Conflict of interest

No potential conflict of interest was reported by the authors.

Thanks

I would to thank the persons that have participated in this work.

IntechOpen

IntechOpen

Author details

Rafik Menassel
Department of Mathematics and Computer Sciences, Larbi Tebessi University,
Tebessa, Algeria

*Address all correspondence to: r.menassel@univ-tebessa.dz

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Galabov M. Fractal image compression. In: Proceedings of 4th International Conference on Computer Systems and Technologies e-Learning – CompSysTech 2003; **43**(6):2003. pp. 320-326
- [2] Selim A, Hadhoud MM, Salem OM. A comparison study between spiral and traditional fractal image compression. In: Proceedings – The 2009 International Conference on Computer Engineering and Systems, ICCES'09. 2009. pp. 39–44
- [3] Thanushkodi KG, Bhavani S. Comparison of fractal coding methods for medical image compression. *IET Image Processing*. 2013; **7**(7):686-693
- [4] Jeng JH, Tseng CC, Hsieh JG. Study on Huber fractal image compression. *IEEE Transactions on Image Processing*. 2009; **18**(5):995-1003
- [5] Li J, Kuo CCJ. Image compression with a hybrid wavelet-fractal coder. *IEEE Transactions on Image Processing*. 1999; **8**(6):868-874
- [6] Han JHJ. Fast fractal image compression using fuzzy classification. In: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 3. 2008. pp. 272-276
- [7] Mitra SK, Murthy CA, Kundu MK. Technique for fractal image compression using genetic algorithm. *IEEE Transactions on Image Processing*. 1998; **7**(4):586-593
- [8] Li J, Yuan D, Xie Q, Zhang C. Fractal image compression by ant colony algorithm. In: 2008 the 9th International Conference for Young Computer Scientists. 2008. pp. 1890-1894
- [9] Tseng C-C, Hsieh J-G, Jeng J-H. Fractal image compression using visual-based particle swarm optimization. *Image and Vision Computing*. 2008; **26**(8):1154-1162
- [10] Lu G. Fractal image compression. *Signal Processing: Image Communication*. 1993; **5**(4):327-343
- [11] Hutchinson J. Fractals and self-similarity. *Indiana University Mathematics Journal*. 1981; **30**(5):713-747
- [12] Barnsley MF, Demko S. Iterated function systems and the global construction of fractals. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. 1817; **1985**(399):243-275
- [13] Peitgen H-O, Jürgens H, Saupe D. *Chaos and fractals*. New Frontiers of Science. New York: Springer Science Business Media; 1992; **1**; ISBN: 978-1-4757-4742-3
- [14] Geir EÖ, Skjalg L. Fractal-based image coding with fast decoder convergence. *Signal Processing*. 1994; **40**(1):105-117
- [15] Ho Moon Y, Soon Kim H, Shin Kim Y, et al. A novel fast fractal decoding algorithm. *Signal Processing: Image Communication*. 1999; **14**(4):325-333. DOI: 10.1016/s0923-5965(98)00016-2
- [16] Moon YH, Baek KR, Kim YS, et al. Fast fractal decoding algorithm with convergence criteria. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*. 1997; **36**(7):1992-1999
- [17] Jacquin AE. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*. 1992; **1**(1):18-30

- [18] Jacquin AE. Fractal image coding: A review. *Proceedings of IEEE*. 1993; **81**(10):1451-1465
- [19] Thomas L, Deravi F. Region-based fractal image compression using heuristic search. *IEEE Transactions on Image Processing*. 1995; **4**(6):832-838
- [20] Cardinal J. Fast fractal compression of greyscale images. *IEEE Transactions on Image Processing*. 2001; **10**(1):159-164
- [21] He C, Xu X, Yang J. Fast fractal image encoding using one-norm of normalized block. *Chaos, Solitons and Fractals*. 2006; **27**(5):1178-1186
- [22] Tong CS, Pi M. Fast fractal image encoding based on adaptive search. *IEEE Transactions on Image Processing*. 2001; **10**(9):1269-1277
- [23] Hartenstein H, Saupe D. Lossless acceleration of fractal image encoding via the fast Fourier transform. *Signal Processing: Image Communication*. 2000; **16**(4):383-394
- [24] Zhang C, Zhou Y, Zhang Z. Fast fractal image encoding based on special image features. *Tsinghua Science and Technology*. 2007; **12**(1):58-62
- [25] Truong TK, Jeng JH, Reed IS, et al. A fast encoding algorithm for fractal image compression using the DCT inner product. *IEEE Transactions on Image Processing*. 2000; **9**(4):529-535
- [26] Lin YL, Wu MS. An edge property-based neighborhood region search strategy for fractal image compression. *Computers & Mathematics with Applications*. 2011; **62**(1):310-318. DOI: 10.1016/j.camwa.2011.05.011
- [27] Olamaei J, Niknam T, Gharehpetian G. Application of particle swarm optimization for distribution feeder reconfiguration considering distributed generators. *Applied Mathematics and Computation*. 2008; **201**(1-2):575-586. DOI: 10.1016/j.amc.2007.12.053
- [28] Gandomi AH, Alavi AH. Multi-stage genetic programming: A new strategy to nonlinear system modeling. *Information Sciences (Ny)*. 2011; **181**(23):5227-5239. DOI: 10.1016/j.ins.2011.07.026
- [29] Blum C, Merkle D. *Swarm intelligence: Introduction and applications*. In: *Natural Computing Series*. Berlin, Heidelberg: Springer-Verlag; 2008. ISBN: 978-3-540-74088-9. DOI: 10.1007/978-3-540-74089-6_2
- [30] Felix T, Chan S, Tiwari MK. *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, (Ed.), ISBN: 978-3-902613-09-7. Rijeka: IntechOpen; 2007
- [31] Wu HS, Zhang FM. Wolf pack algorithm for unconstrained global optimization. In: *Mathematical Problems in Engineering*. Hindawi Publishing Corporation. Vol. 2014. 2014. Available from: <http://dx.doi.org/10.1155/2014/465082>. Article ID: 465082
- [32] Yang XS. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N, editors. *Nature Inspired Cooperative Strategies for Optimization (NICSO)*. *Studies in Computational Intelligence*. Vol. 284. Berlin, Heidelberg: Springer; 2010. pp. 65-74. DOI: 10.1007/978-3-642-12538-6_6k2
- [33] Karaboga D. A comparative study of artificial bee Colony algorithm. *Applied Mathematics and Computation*. 2009; **214**(1):108-132. DOI: 10.1016/j.amc.2009. 03.090
- [34] Martinez C. An ACO algorithm for image compression. *CLEI Electronic Journal*. 2006; **9**(2):1-17. DOI: 10.19153/cleiej.9.2.1

- [35] Chakrapani Y, Soundararajan K. Genetic algorithm applied to fractal image compression. *ARPN Journal of Engineering and Applied Sciences*. 2009;4(1):53-58
- [36] Xing-yuan W, Fan-ping L, Shu-guo W. Fractal image compression based on spatial correlation and hybrid genetic algorithm. *Journal of Visual Communication and Image Representation*. 2009;20(8):505-510. DOI: 10.1016/j.jvcir.2009.07.002
- [37] Chakrapani Y, Soundararajan K. Implementation of fractal image compression employing particle swarm optimization. *The World Journal of Modelling and Simulation*. 2010;6(1):40-46
- [38] Al-Bundi SS, Al-Saidi NMG, Al-Jawari NJ. Crowding optimization method to improve fractal image compressions based iterated function systems. *International Journal of Advanced Computer Science and Applications*. 2016;7(7):392-401
- [39] Al-Saidi NMG, Al-Bundi SS, Al-Jawari NJ. An improved harmony search algorithm for reducing computational time of fractal image coding. *Journal of Theoretical and Applied Information Technology*. 2017;95(8):1669-1678
- [40] Menassel R, Nini B, Mekhaznia T. An improved fractal image compression using wolf pack algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*. 2018;30(3):429-439. DOI: 10.1080/0952813X.2017.1409281
- [41] Menassel R, Gaba I, Titi K. Introducing BAT inspired algorithm to improve fractal image compression. *International Journal of Computers and Applications*. 2019. DOI: 10.1080/1206212X.2019.1638631
- [42] Kalunge VV, Varunakshi B. Time optimization of fractal image compression by using genetic algorithm. *International Journal of Engineering Research & Technology*. 2012;1(10):1-5
- [43] Wang X-Y, Zhang D-D, Wei N. Fractal image coding algorithm using particle swarm optimization and hybrid Quadtree partition scheme. *IET Image Processing*. 2015;9(2):153-161
- [44] Wang X-Y, Zhang D-D. Discrete wavelet transform-based simple range classification strategies for fractal image coding. *Nonlinear Dynamics*. 2014;75(3):439-448