Bat Algorithm for Kernel Computation in Fractal Image Reconstruction

Akemi Gálvez^{1,2}, Eneko Osaba³, Javier Del Ser^{3,4}, Andrés Iglesias^{1,2,*}

¹Department of Information Science, Faculty of Sciences, Narashino Campus Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan

²Dpt. of Applied Mathematics & Comp. Sci., E.T.S.I. Caminos, Canales y Puertos

Universidad de Cantabria, Avda. de los Castros, s/n, 39005, Santander, Spain

³TECNALIA Research & Innovation, 48160, Derio, Spain

⁴University of the Basque Country (UPV/EHU), Bilbao, Spain *Corresponding author. Email: iglesias@unican.es

Abstract. Computer reconstruction of digital images is an important problem in many areas such as image processing, computer vision, medical imaging, sensor systems, robotics, and many others. A very popular approach in that regard is the use of different kernels for various morphological image processing operations such as dilation, erosion, blurring, sharpening, and so on. In this paper, we extend this idea to the reconstruction of digital fractal images. Our proposal is based on a new affine kernel particularly tailored for fractal images. The kernel computes the difference between the source and the reconstructed fractal images, leading to a difficult nonlinear constrained continuous optimization problem, solved by using a powerful nature-inspired metaheuristics for global optimization called the bat algorithm. An illustrative example is used to analyze the performance of this approach. Our experiments show that the method performs quite well but there is also room for further improvement. We conclude that this approach is promising and that it could be a very useful technique for efficient fractal image reconstruction.

Keywords: image processing \cdot image reconstruction \cdot affine kernel \cdot fractal image \cdot bat algorithm.

1 Introduction

Computer reconstruction of digital images is a classical problem in fields such as image processing and computer vision. The topic has gained strong relevance during the last few decades owing to its important applications in several areas, including medical imaging (computer tomography, magnetic resonance), sensor systems, robotics, smart cities, internet of things, and many others. Roughly speaking, the problem consists of reproducing a given image described in terms of digital data (typically, raster or bitmapped images) by following procedures involving either a set of equations and operators, or a set of rules, or some kind of heuristics (even sometimes combinations of them). In this paper, we are interested in this problem for the case of fractal images, which exhibit a property

A. Gálvez, E. Osaba, J. Del Ser, A. Iglesias

 $\mathbf{2}$

called *self-similarity*, meaning that the images follow (at least, approximately) a self-similar pattern across different scales [2, 4].

Several methods have been traditionally applied to the image reconstruction problem. When dealing with fractal images, some popular methods include the Brownian motion, escape-time fractals, finite subdivision rules, L-systems, strange attractors of dynamical systems [11], and many others [2, 8, 13]. For a general (non-fractal) image, other methods based on image processing techniques are more commonly applied [10]. Among them, a popular approach in image processing is the use of different kernels for various morphological image processing operations such as dilation, erosion, blurring, sharpening, and so on. In this work, we are interested to follow this approach regarding its potential application to the case of fractal images.

In this paper, we introduce a new method for digital fractal image reconstruction. Our proposal is based on a new affine kernel particularly tailored for fractal images. The kernel computes the difference between the source and the reconstructed fractal images, according to a given metrics. This leads to a difficult nonlinear constrained continuous optimization problem that has been proved to be not well suited for classical mathematical optimization techniques. To tackle this issue, we make use of a powerful nature-inspired metaheuristics for global optimization called bat algorithm (see Section 3 for details).

The structure of this paper is as follows: Section 2 summarizes the mathematical background required to follow the paper. Section 3 describes the main features of the bat algorithm, the global optimization metaheuristics used in this paper. Our proposed method is described in detail in Section 4 and then applied to an illustrative example in Section 5. The paper closes with the conclusions and some ideas for future work.

2 Basic Concepts and Definitions

2.1 Digital Images

In this work, we consider a *digital image* \mathcal{I} to be numerically represented as a twodimensional raster or bitmapped image. We exclude in our study other possible computer representations such as vector images. The *convolution operator* of two functions ϕ and ψ , denoted by $\phi \otimes \psi$, is a mathematical operation describing how the shape of one function is modified by the other. Analytically, it is given by an integral transform of both functions defined as:

$$(\phi \otimes \psi)(\rho) = \int_{-\infty}^{\infty} \phi(\tau)\psi(\rho - \tau)d\tau$$
(1)

In the context of image processing, the convolution operator is carried out in a discrete fashion by using a kernel applied on a given image \mathcal{I} via matrix convolution. Let \mathcal{K} be such a kernel. The convolution is given by:

$$\mathcal{I}'_{x,y} = (\mathcal{K} \otimes \mathcal{I})_{x,y} = \sum_{\alpha = -\mu}^{\mu} \sum_{\beta = -\nu}^{\nu} \mathcal{K}_{\alpha,\beta} \mathcal{I}_{x-\alpha,y-\beta}$$
(2)

where \mathcal{I}' is the transformed image, the subscripts indicate the image pixels, and $\mathcal{K} = \{\mathcal{K}_{\alpha,\beta}\}_{\alpha,\beta}$, with $-\mu \leq \alpha \leq \mu, \nu \leq \beta \leq \nu$. Depending on the particular purposes, different kernels can be considered: for instance, classical operations in morphological image processing such as dilation and erosion are expressed by specific filtering kernels operating on a input binary image. Other operations such as opening, closing, and boundary detection can be obtained as a combination of such kernels (see [10] for details).

2.2 Fractal Images

In this paper, a digital fractal image is defined as a digital image with the property of self-similarity and whose fractal dimension is larger than its topological dimension [4, 12]. Suppose a set of affine mappings $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_\eta\}$ defined on a complete metric space $\mathcal{M} = (\mathbf{\Omega}, \Psi)$, where $\mathbf{\Omega} \subset \mathbb{R}^n$ and Ψ is a distance on $\mathbf{\Omega}$. Such affine mappings Λ_{κ} can be represented by a 3×3 augmented matrix $\mathbf{\Theta}_{\kappa} = \{\theta_{i,j}^{\kappa}\}_{i,j=1,2,3}$ in homogeneous coordinates, with: $\theta_{3,j}^{\kappa} = \delta_{3,j}$, where δ represents the Kronecker delta. In that case, $\Lambda_{\kappa}(A) = \mathbf{\Theta}_{\kappa}.A^*, \forall A \subset \mathbb{R}^2$, where the superscript * denotes the augmented matrix. We assume that all mappings Λ_{κ} are contractive, with contractivity factor $\lambda_{\kappa} > 0$.

Consider now the set of all compact subsets of the plane, \mathcal{H} . We can define the Hutchinson operator, Ξ as:

$$\Xi(S) = \bigcup_{\kappa=1}^{\eta} \Lambda_{\kappa}(S) \tag{3}$$

for each $S \in \mathcal{H}$. Since all Λ_{κ} are contractions, this operator Ξ is also a contraction in \mathcal{H} with the induced Hausdorff metric [3, 14]. Then, according to the fixed point theorem, Ξ has a unique fixed point, called the *attractor of* Λ .

The reconstruction of digital fractal images is driven by a famous result by M. Barnsley called the Collage Theorem [2]. Roughly speaking, it states that every digital image can be represented as the attractor of a system Λ . In particular, given a non-empty $B \in \mathcal{H}$, the induced Hausdorff metric H(.,.) on \mathcal{H} , a non-negative real threshold value $\epsilon \ge 0$, and a system of affine contractive mappings Λ with contractivity factor $0 < \lambda < 1$, given by: $\lambda = \max{\{\lambda_{\kappa}\}_{\kappa=1,...,\eta}}$, if $H(B, \Xi(B)) \le \epsilon$ then $H(B, \mathcal{A}) \le \frac{\epsilon}{1-\lambda}$, where \mathcal{A} is the attractor of Λ , or equivalently: $H(B, \mathcal{A}) \le \frac{1}{1-\lambda} H\left(B, \bigcup_{\kappa=1}^{\eta} \phi_{\kappa}(B)\right)$.

3 The Bat Algorithm

The *bat algorithm* is a bio-inspired swarm intelligence algorithm originally proposed by Xin-She Yang in 2010 to solve continuous optimization problems [21–23]. The algorithm is based on the echolocation behavior of microbats, which use a type of sonar called *echolocation*, with varying pulse rates of emission and

loudness, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. The idealization of the echolocation of microbats is as follows:

- 1. Bats use echolocation to sense distance and distinguish between food, prey and background barriers.
- 2. Each virtual bat flies randomly with a velocity \mathbf{v}_i at position (solution) \mathbf{x}_i with a fixed frequency f_{min} , varying wavelength λ and loudness A_0 to search for prey. As it searches and finds its prey, it changes wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r, depending on the proximity of the target.
- 3. It is assumed that the loudness will vary from an (initially large and positive) value A_0 to a minimum constant value A_{min} .

Some additional assumptions are advisable for further efficiency. For instance, we assume that the frequency f evolves on a bounded interval $[f_{min}, f_{max}]$. This means that the wavelength λ is also bounded, because f and λ are related to each other by the fact that the product λ . f is constant. For practical reasons, it is also convenient that the largest wavelength is chosen such that it is comparable to the size of the domain of interest (the search space for optimization problems). For simplicity, we can assume that $f_{min} = 0$, so $f \in [0, f_{max}]$. The rate of pulse can simply be in the range $r \in [0, 1]$, where 0 means no pulses at all, and 1 means the maximum rate of pulse emission.

With these idealized rules indicated above, the basic pseudo-code of the bat algorithm is shown in Algorithm 1. Basically, the algorithm considers an initial population of \mathcal{P} individuals (bats). Each bat, representing a potential solution of the optimization problem, has a location \mathbf{x}_i and velocity \mathbf{v}_i . The algorithm initializes these variables with random values within the search space. Then, the pulse frequency, pulse rate, and loudness are computed for each individual bat. Then, the swarm evolves in a discrete way over iterations, like time instances until the maximum number of iterations, \mathcal{G}_{max} , is reached. For each generation g and each bat, new frequency, location and velocity are computed according to the following evolution equations:

$$f_i^g = f_{min}^g + \beta (f_{max}^g - f_{min}^g) \tag{4}$$

$$\mathbf{v}_i^g = \mathbf{v}_i^{g-1} + \left[\mathbf{x}_i^{g-1} - \mathbf{x}^*\right] f_i^g \tag{5}$$

$$\mathbf{x}_i^g = \mathbf{x}_i^{g-1} + \mathbf{v}_i^g \tag{6}$$

where $\beta \in [0, 1]$ follows the random uniform distribution, and \mathbf{x}^* represents the current global best location (solution), which is obtained through evaluation of the objective function at all bats and ranking of their fitness values. The superscript $(.)^g$ is used to denote the current generation g. The best current solution and a local solution around it are probabilistically selected according to some given criteria. Then, search is intensified by a local random walk. For this local search, once a solution is selected among the current best solutions, it is perturbed locally through a random walk of the form: $\mathbf{x}_{new} = \mathbf{x}_{old} + \epsilon \mathcal{A}^g$, where ϵ is a uniform random number on [-1, 1] and $\mathcal{A}^g = \langle \mathcal{A}^g_i \rangle$, is the average

Require: (Initial Parameters)

Population size: \mathcal{P} ; Maximum number of iterations: \mathcal{G}_{max} ; Loudness: \mathcal{A} Pulse rate: r; Maximum frequency: f_{max} ; Dimension of the problem: dObjective function: $\phi(\mathbf{x})$, with $\mathbf{x} = (x_1, \dots, x_d)^T$; Random number: $\theta \in U(0, 1)$ 1: $g \leftarrow 0$ 2: Initialize the bat population \mathbf{x}_i and \mathbf{v}_i , (i = 1, ..., n)3: Define pulse frequency f_i at \mathbf{x}_i 4: Initialize pulse rates r_i and loudness \mathcal{A}_i 5: while $g < \mathcal{G}_{max}$ do 6: for i = 1 to \mathcal{P} do Generate new solutions by using eqns. (4)-(6)7: 8: if $\theta > r_i$ then $\mathbf{s}^{best} \leftarrow \mathbf{s}^{g}$ 9: //select the best current solution $\mathbf{ls}^{best} \leftarrow \mathbf{ls}^{g}$ //generate a local solution around \mathbf{s}^{best} 10: 11: end if Generate a new solution by local random walk 12:if $\theta < A_i$ and $\phi(\mathbf{x_i}) < \phi(\mathbf{x^*})$ then 13:Accept new solutions, increase r_i and decrease \mathcal{A}_i 14:15:end if 16:end for 17: $q \leftarrow q + 1$ 18: end while 19: Rank the bats and find current best \mathbf{x}^* 20: return \mathbf{x}^*

Algorithm 1: Bat algorithm pseudocode

loudness of all the bats at generation g. If the new solution achieved is better than the previous best one, it is probabilistically accepted depending on the value of the loudness. In that case, the algorithm increases the pulse rate and decreases the loudness. This process is repeated for the given number of iterations. In general, the loudness decreases once a new best solution is found, while the rate of pulse emission decreases. For simplicity, the following values are commonly used: $\mathcal{A}_0 = 1$ and $\mathcal{A}_{min} = 0$, assuming that this latter value means that a bat has found the prey and temporarily stop emitting any sound. The evolution rules for loudness and pulse rate are as: $\mathcal{A}_i^{g+1} = \alpha \mathcal{A}_i^g$ and $r_i^{g+1} = r_i^0 [1 - exp(-\gamma g)]$ where α and γ are constants. Note that for any $0 < \alpha < 1$ and any $\gamma > 0$ we have: $\mathcal{A}_i^g \to 0, r_i^g \to r_i^0$ as $g \to \infty$. Generally, each bat should have different values for loudness and pulse emission rate, which can be achieved by randomization. To this aim, we can take an initial loudness $\mathcal{A}_i^0 \in (0, 2)$ while the initial emission rate r_i^0 can be any value in the interval [0, 1]. Loudness and emission rates will be updated only if the new solutions are improved, an indication that the bats are moving towards the optimal solution.

Bat algorithm is a very promising method that has already been successfully applied to several problems, such as multilevel image thresholding [1], economic dispatch [18], B-spline curve reconstruction [15], optimal design of structures in

civil engineering [17], robotics [20], fuel arrangement optimization [16], planning of sport training sessions [5], transport [19], and many others. The interested reader is also referred to the general paper in [24] for a comprehensive review of the bat algorithm, its variants and other interesting applications.

4 The Method

4.1 Optimization Problem

Suppose that we are given a digital fractal image, \mathcal{I} . The Collage Theorem states that \mathcal{I} can be closely approximated by an iterative process driven by a set of contractive affine mappings, $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_\eta\}$, on the two-dimensional real plane. In particular, for any arbitrary $S_0 \in \mathcal{H}$, consider $S_j = \Lambda_{\kappa}(S_{j-1}) = \mathbf{\Theta}_{\kappa}.S_{j-1}$, where κ is randomly chosen from the set of indices $\{1, \ldots, \eta\}$ according to a set of probabilities $\mathcal{W} = \{\omega_1, \ldots, \omega_\eta\}$, with $\sum_{\kappa=1}^{\eta} \omega_{\kappa} = 1$, for each iteration step j. Then, the sequence $\{S_j\}_j$ converges to \mathcal{I} as $j \to \infty$.

In other words, any given digital fractal image \mathcal{I} can be accurately approximated by the action of a finite collection of affine kernels $\{\Theta_{\kappa}\}_{\kappa=1,...,\eta}$ according to a similarity function \mathcal{S} , which measures the graphical distance between \mathcal{I} and the reconstructed image $\mathcal{I}' = \bigcup_{\kappa=1}^{\eta} \Lambda_{\kappa}(\mathcal{I})$. In line with this, the problem consists of computing the kernels Θ_{κ} and can be formulated as the following optimization problem:

$$\underset{\{\Theta_{i,j}^{\kappa}\},\{\omega_{\kappa}\}}{\text{minimize }} \mathcal{S}\left(\mathcal{I},\bigcup_{\kappa=1}^{\eta}\Lambda_{\kappa}(\mathcal{I})\right)$$
(7)

The minimization in Eq. (7) is a continuous nonlinear constrained optimization problem, because all free variables $\{\Theta_{i,j}^{\kappa}\}_{i,j,\kappa}, \{\omega_{\kappa}\}_{\kappa}$ are real-valued and must satisfy the condition that the corresponding functions Λ_{κ} have to be contractive. It is also a multimodal problem, as there can be several global or local minima of the similarity function. The problem is so difficult that only partial solutions have been reported so far in the literature. However, the general problem still remains unsolved. In this paper we address this problem by applying the bat algorithm described in previous section.

4.2 The Procedure

In our method, we consider an initial population of χ individuals called bats, $\{\mathcal{B}_i^0\}_{i=1,\ldots,\chi}$, where each bat is a real-valued vector comprised of all free variables in Eq. (7) and the superscrit denotes the iteration number. These individuals are initialized with uniform random values in [-1,1] for the variables in $\{\mathcal{O}_{i,j}^{\kappa}\}_{i,j,\kappa}$, and in [0,1] for the $\{\omega_{\kappa}\}_{\kappa}$, such that $\sum_{\kappa=1}^{\eta} \omega_{\kappa}^i = 1$. After this initialization step, we compute the contractive factors λ_{κ} and reinitialize all functions Λ_{κ} with $\lambda_{\kappa} \geq 1$ to ensure that only contractive functions are included in the initial population. Regarding the fitness function, it is given by the Hamming distance:



Fig. 1. Six different individuals (bats) from the initial random population.

8



Fig. 2. (l-r, t-b) Evolution of the global best of the population from 100 to 600 iterations with step size 100, respectively.

the fractal images are stored as binary bitmap images for a given resolution defined by a mesh size parameter, m_s . Then, we divide the number of mismatches between the original and the reconstructed matrices by the total number of boxes in the image. This yields the *normalized similarity error rate index* (NSERI)



Fig. 2. (cont'd) (l-r, t-b) Evolution of the global best of the population from 800 to 1400 iterations with step size 200, respectively.

between both images, denoted by $|\mathcal{S}(\mathcal{I}, \mathcal{I}')|$. This is the fitness function used in this work.

4.3 Parameter Tuning

A critical issue when working with swarm intelligence techniques is the parameter tuning, which is well-known to be problem-dependent. Our choice has been fully empirical, based on computer simulations for different parameter values. The different parameters used in this work are arranged in rows in Table 1. For each parameter, the table shows (in columns) its symbol, meaning, range of values, and the parameter value chosen in this paper. Regarding the stopping criterion, our method is run for a fixed number of iterations, \mathcal{G}_{max} . From our experiments, we found that $\mathcal{G}_{max} = 2500$ iterations is enough to reach convergence in all our simulations, so this is the value used in this work. Finally, our method requires to define the mesh size, m_s , set to $m_s = 100$ in this work.

With this choice of parameter values, we run the bat algorithm iteratively. Positions and velocities of the bats are computed according to the evolution

10 A. Gálvez, E. Osaba, J. Del Ser, A. Iglesias

Symbol	Meaning	Range of Values	Selected Value
\mathcal{P}	population size	50 - 200	100
\mathcal{G}_{max}	maximum number of iterations	1000-5000	2500
\mathcal{A}^0	initial loudness	(0, 2)	0.5
\mathcal{A}_{min}	minimum loudness	[0, 1]	0
r^0	initial pulse rate	[0, 1]	0.2
f_{max}	maximum frequency	[0, 10]	1.5
α	multiplicative factor	(0, 1)	0.3
γ	exponential factor	[0, 1]	0.2

Table 1. Bat algorithm parameters and their values in this paper.

equations (4)-(6) and then ranked according to the fitness function explained above. This iterative process stops when the maximum number of iterations \mathcal{G}_{max} is reached. The best solution achieved at the final iteration is taken as the solution of the optimization problem.

5 An Illustrative Example

5.1 Graphical Results

Our method has been applied to several examples. However, we restrict our discussion in this paper to just one illustrative example because of limitations of space. In the example, the original image, shown in Fig. 3(top), is reconstructed with three affine transformations Λ_{κ} , $\kappa = 1, 2, 3$. We apply our method by using an initial population of randomly chosen 100 bats. For illustration, six of them are displayed in Fig. 1. As the reader can see, they are visually very different to each other, and all them are very far from the original source image. Then, our method is applied for $\mathcal{G}_{max} = 2500$ iterations as described above.

Fig. 2 shows the evolution of the global best of the population at specific iteration values, ranging from 100 to 600 with step size 100, and then from 800 to 1400 with step size 200. From the picture, we can see that the global best is very far from the source image at initial stages of the method, leading to images that do not really resemble the goal image. However, as the number of iterations increases, the global best image is getting visually closer to the intended image. Also, note that the variation of the global shape of the image over the iterations is more dramatic at initial stages, corresponding to a higher explorative phase, while it varies slightly at later iterations, where the image approaches to the target image by small incremental improvements of some local features, corresponding to the exploitative phase of the method. Fig. 3 (bottom) shows the reconstructed image after the convergence is reached. From Fig. 3 we can see that the final reconstructed image is very similar visually to the source image, capturing faithfully all major features of a very complicated and irregular shape. This means that our method is able to reconstruct the general shape of



Fig. 3. (top) Original image; (bottom) best reconstructed image.

the given image with a high visual accuracy. The corresponding convergence diagram is shown in Fig. 4.



Fig. 4. Convergence diagram of the NSERI fitting error for 2500 iterations.

5.2 Numerical Results

Regarding the numerical results, the similarity error between the original and the reconstructed images is 0.4748 according to our metric, meaning that we got a 47% of mismatches between both images for the given resolution. This result may seem surprising in the light of the good visual results, but it must be taken into account that our metrics computes the differences based on the numerical values on the grid. Therefore, any minor distortion of the image (e.g., displacement, rotation, or scaling) can yield substantial increases in the similarity error, even though the general shape might still be well replicated. Furthermore, even if these variations happen at a local level, they have a dramatic effect on the numerical results. Of course, this effect can be partially alleviated by considering a less demanding fitness function. However, we preferred to preserve this more stringent metric in order to push our method further looking for a higher accuracy. As a conclusion, in spite of the good graphical results, the numerical results show that the method is not optimal yet and there is probably room for further improvement.

5.3 Computational Issues

All computations in this paper have been performed on a 2.6 GHz Intel Core i7 processor with 16 GB of RAM. The source code has been implemented by the authors in the native programming language of the popular scientific program

Matlab version 2015a using the numerical libraries for fractals in [6, 7, 9]. Regarding the CPU times, they depend on the complexity of the image, the resolution of the mesh, and other factors. For illustration, each single execution takes about $25\sim30$ minutes. In general, we noticed that the method is time-consuming for very high resolution images. This is the case for the image in our example, which is drawn with 5×10^5 points.

6 Conclusions and Future Work

This paper introduces a new approach for digital fractal image reconstruction. The method is based on a new affine kernel inspired by those in morphological image processing but specifically designed for fractal images. This approach leads to a difficult multimodal nonlinear continuous optimization, solved by using a powerful nature-inspired metaheuristics: the bat algorithm. An illustrative example is used to analyze the performance of this approach. Our experiments show that the method obtains very good visual results. However, the numerical results are not optimal yet, suggesting that there is also room for further improvement. We conclude that this approach is promising and it could potentially become (after further improvement to reduce the computing times and enhance its numerical accuracy) a very useful technique in the context of fractal image reconstruction.

Regarding our future work, we want to modify our method to improve our numerical and graphical results. In addition to a more optimized fitness function, we are interested to hybridize the bat algorithm with local search procedures to enhance the exploitation abilities of the method in the neighborhood of the local optima for higher accuracy. We also wish to extend our results to the cases of non-binary and colored images, with the possible addition of an extra color channel. Reducing our CPU times for better performance is also part of our plans for future work in the field.

Acknowledgments

Akemi Gálvez and Andrés Iglesias acknowledge the financial support from the project PDE-GIR of the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 778035, and from the Spanish Ministry of Science, Innovation and Universities (Computer Science National Program) under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE). Eneko Osaba and Javier Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program.

References

1. Alihodzic, A., Tuba, M.: Improved bat algorithm applied to multilevel image thresholding. *The Scientific World Journal*, **2014**, article ID 176718, 16 pages (2014).

- 14 A. Gálvez, E. Osaba, J. Del Ser, A. Iglesias
- 2. Barnsley, M.F.: Fractals Everywhere (2nd Ed.). Academic Press, San Diego (1993).
- Elton, J.H.: An Ergodic Theorem for Iterated Maps. Ergodic Theory Dynam. Syst., 7, 481–488 (1987).
- Falconer, K.: Fractal Geometry: Mathematical Foundations and Applications (2nd Ed.). John Wiley & Sons, Chichester, England (2003).
- Fister, I., Rauter, S., Yang, X.-S., Ljubic, K., Fister Jr., I.: Planning the sports training sessions with the bat algorithm. *Neurocomputing*, **149**, Part B, 993–1002 (2015).
- Gálvez, A.: IFS Matlab generator: a computer tool for displaying IFS fractals. In: Proc. ICCSA'2009, pp. 132–142. IEEE CS Press, Los Alamitos, CA (2009).
- Gálvez, A., Iglesias, A., Takato, S.: Matlab-based KETpic add-on for generating and rendering IFS fractals. CCIS, 56, 334–341 (2009).
- Gálvez, A., Iglesias, A., Takato, S.: KETpic Matlab binding for efficient handling of fractal images. Int. J. Future Generation Comm. & Networking, 3(2), 1–14 (2010).
- Gálvez, A., Kitahara, K., Kaneko, M.: IFSGen4LaTeX: Interactive graphical user interface for generation and visualization of iterated function systems in L^AT_EX. LNCS 8592, 554–561 (2014).
- Gonzalez, R.C., Woods, R.E.: Digital Image Processing. 4th Edition. Pearson Prentice Hall, N.J. (2017).
- 11. Gutiérrez, J.M., Iglesias, A.: A Mathematica package for the analysis and control of chaos in nonlinear systems. *Computers in Physics*, **12**(6), 608–619 (1998).
- Gutiérrez, J.M., Iglesias, A., Rodríguez, M.A.: A multifractal analysis of IFSP invariant measures with application to fractal image generation. *Fractals* 4(1), 17– 27 (1996).
- Gutiérrez, J.M., Iglesias, A., Rodríguez, M.A., Rodríguez, V.J.: Generating and rendering fractal images. *The Mathematica Journal*, 7(1), 6–13 (1997).
- Hutchinson, J.E.: Fractals and self similarity. Indiana Univ. Math. Journal, 30(5), 713–747 (1981).
- Iglesias, A., Gálvez, A., Collantes, M.: Multilayer embedded bat algorithm for Bspline curve reconstruction. *Integ. Computer-Aided Eng.*, 24(4), 385–399 (2017).
- Kashi, S., Minuchehr, A., Poursalehi, N., Zolfaghari, A.: Bat algorithm for the fuel arrangement optimization of reactor core. *Annals of Nuclear Energy*, 64, 144–151 (2014).
- Kaveh, A., Zakian, P.: Enhanced bat algorithm for optimal design of skeletal structures. Asian Journal Civil Engineering, 15(2), 179–212 (2014).
- Latif, A., Palensky, P.: Economic dispatch using modified bat algorithm. Algorithms, 7(3), 328–338 (2014).
- Osaba, E., Yang, X.S. Diaz, F., Lopez-Garcia, P., Carballedo, R.: An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Eng. Appl. of Artificial Intelligence* 48, 59–71 (2016).
- Suárez, P., Iglesias, A., Gálvez, A.: Make robots be bats: specializing robotic swarms to the bat algorithm. Swarm and Evolutionary Computation, 44(1), 113–129 (2019).
- Yang, X.S.: A new metaheuristic bat-inspired algorithm. Studies in Computational Intelligence, 284, 65–74 (2010).
- Yang, X. S.: Bat algorithm for multiobjective optimization. Int. J. Bio-Inspired Computation, 3(5), 267–274 (2011).
- Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, 29(5), 464–483 (2012).
- Yang, X.S.: Bat algorithm: literature review and applications. Int. J. Bio-Inspired Computation, 5(3), 141–149 (2013).