Fast Fractal Image Decoding Based on Optimal Iterated Function System

Authors: Qiang Wang, Qiang Wang

Date: 2025-02-25T00:00:00+00:00

Abstract

An optimal iterated function system (OIFS) based fast fractal decoding method was proposed in this study. Based on the minimum domain block set (MDBS) which can effectively remove redundant domain blocks and accelerate the decoding process, we found that there still exist redundant domain blocks in MDBS, and the decoding process can be accelerated further. By introducing the definition of minimum iterated function system (MIFS) and multilevel MIFSs, the redundant domain blocks can be removed level by level, and finally the optimal iterated function system (OIFS) can be obtained. Before the first iteration at decoding phase, the OIFS can be firstly established by removing the redundant domain blocks outside OIFS. Then, only the range blocks inside OIFS were reconstructed from the first to penultimate iterations, and the computations of reconstructing the range blocks outside OIFS can be saved. Finally, the whole image was reconstructed in the last iteration. Three fractal coding methods were adopted to assess the performance of the proposed method. Experimental results show that about 1%-10% of total computations in decoding process can be further saved regarding the MDBS based method.

Full Text

Abstract

This study proposes a fast fractal decoding method based on an optimal iterated function system (OIFS). Building upon the minimum domain block set (MDBS), which effectively removes redundant domain blocks to accelerate decoding, we discovered that redundant domain blocks persist even within MDBS, offering further opportunities for acceleration. By introducing the concept of minimum iterated function system (MIFS) and multilevel MIFSs, redundant domain blocks can be eliminated hierarchically, ultimately yielding the optimal iterated function system (OIFS). During decoding, OIFS is established before the first iteration by removing redundant domain blocks outside its scope.

Subsequently, only range blocks within OIFS are reconstructed from the first through penultimate iterations, saving computations for reconstructing blocks outside OIFS. The complete image is reconstructed in the final iteration. We evaluated the proposed method using three fractal coding approaches. Experimental results demonstrate that the method saves approximately 1%-10% of total computations compared to the MDBS-based approach.

Keywords: Fractal image coding; Fast fractal image decoding; Optimal iterated function system

1. Introduction

Fractal image coding represents a successful application of fractal theory to image compression. Nearly three decades ago, following Barnsley' s introduction of the core concept of fractal image coding [?], Jacquin proposed the first practical fractal coding algorithm, which has since been recognized as a promising image compression technique [?]. Fractal image coding offers several advantages, including innovative principles, potential for high compression ratios, fast decoding, and resolution independence. However, it suffers from high computational complexity during encoding. To address this limitation and enhance practical applicability, researchers worldwide have developed various fast fractal encoding methods. Some have proposed local block matching-based approaches [?, ?], which accelerate encoding by converting exhaustive block matching into local block matching within the domain block pool while preserving decoded image quality. To achieve further speedup, others have introduced no-search fractal image coding methods [?]. These techniques directly assign the best-matched domain block without block matching operations, enabling real-time encoding at the cost of reduced image quality. In recent years, significant improvements have been made in other aspects of fractal image coding. For instance, block-matching operations can be simplified using fast affine transformations to accelerate encoding [?, ?], while linear affine transformations can be replaced with nonlinear ones to improve decoded image quality [?]. Several hybrid coding methods have also been proposed to enhance overall performance [?, ?]. After three decades of development, fractal image coding has found applications in diverse image processing domains, including image retrieval [?], watermarking [?], image magnification [?], medical image processing [?], image hashing [?, ?], human pose estimation [?, ?], image denoising [?], and image encryption [?].

Although fractal image decoding typically converges in a small number of iterations, accelerating the decoding process remains valuable for real-time applications. To expedite fractal decoding, some researchers employ initial images that closely approximate the input image, enabling faster convergence to the final decoded image through specified iteration methods [?]. Others adopt improved iteration strategies that effectively reduce iteration errors relative to the final decoded image, thereby accelerating the process [?, ?]. Recently, a fundamentally different approach based on minimum domain block set (MDBS) was proposed to reduce computational complexity during decoding [?]. MDBS contains only the essential domain blocks required to reconstruct range blocks both inside and outside the set. Consequently, during decoding, only range blocks within MDBS are reconstructed in each iteration from the first to the penultimate, saving computations for blocks outside MDBS and speeding up the process.

In this study, we identified redundant domain blocks even within MDBS, indicating potential for further acceleration. We first introduce the concept of minimum iterated function system (MIFS) and multilevel MIFSs. Redundant domain blocks can then be removed hierarchically, with the MIFS at the final level defined as the optimal iterated function system (OIFS). Before the first decoding iteration, OIFS is established. From the first through penultimate iterations, only range blocks within OIFS are reconstructed, saving computations for blocks outside OIFS. In the final iteration, the local image within OIFS converges, and range blocks outside OIFS are also reconstructed. Compared to the MDBS-based fast decoding method, the proposed approach saves computations for reconstructing range blocks outside OIFS (but inside MDBS) from the first to penultimate iterations, further accelerating fractal decoding. We combined the proposed method with existing fast fractal decoding techniques and evaluated it using Jacquin' s method and two state-of-the-art approaches. Experimental results confirm that the proposed method effectively reduces computational requirements compared to previous methods.

The contributions of this study are:

1. Based on the definition of minimum iterated function system (MIFS), we identified redundant domain blocks within MIFS and introduced multilevel MIFSs. By hierarchically removing redundant domain blocks, we obtain the optimal iterated function system (OIFS).

2. The proposed method can be combined with existing fractal coding methods to accelerate their decoding processes, making them more suitable for real-time applications.

This paper is organized as follows: Section 2 reviews conventional fractal image coding. Section 3 introduces the definitions of MIFS, multilevel MIFSs, and OIFS, and describes the OIFS-based fast decoding method. Sections 4 and 5 present and analyze experimental results, respectively. Section 6 concludes the paper.

2. Conventional Fractal Image Coding

2.1 Traditional Fractal Image Coding

Traditional fractal image coding aims to establish an iterated function system (IFS) whose fixed point approximates the input image. During encoding, the input image is partitioned into range blocks of size $B \times B$, where $i = 1, 2, 3, ..., \text{NumR}$. The domain block pool, D_j, where $j = 1, 2, 3, ..., \text{NumD}$, is obtained by sliding a $2B \times 2B$ window over the input image. For each range block R_i, the best-matched domain block is found through exhaustive search within the domain block pool, which involves performing eight isometric transformations. The collage error for an arbitrary range block is computed as:

$$R_i = \arg\min_{D_j,\varphi_{ij}} \|R_i - \varphi_{ij}(D_j)\|^2 = \arg\min_{\alpha_i,\beta_i,\gamma_i} \|R_i - (\alpha_i \cdot \eta(\gamma_i(D_j)) + \beta_i \cdot I)\|^2$$

where $i = 1, 2, 3, ..., \text{NumR}$ and $j = 1, 2, 3, ..., \text{NumD}$. Here, $\varphi_{ij}(\cdot)$ denotes the mapping operation from D_j to R_i, $\eta(\cdot)$ and $\gamma(\cdot)$ represent contracting and isometric transformations, respectively, and α_i and β_i denote the scaling and offset coefficients of the affine transformation. I denotes a $B \times B$ block with all components equal to one. The encoding process establishes mapping operations from domain blocks to range blocks. All domain and range blocks, along with their associated mapping operations, constitute an IFS:

$$\text{IFS} : \{R_i, D_j, \varphi_{ij}\}, \quad i = 1, 2, 3, ..., \text{NumR}, \quad j = 1, 2, 3, ..., \text{NumD}$$

During decoding, any $M \times N$ image can serve as the initial image. The decoding process converges to the final decoded image after several iterations using the same IFS from encoding. If the Airplane image was encoded, Figures 1(b)-(f) illustrate the first five iteration images when the rotated Airplane image in Fig. 1 Figure 1: see original paper is used as the initial image. The iteration images gradually converge to the decoded image, which approximates the input image well.

2.2 Fast Fractal Image Decoding

Existing fast fractal decoding methods fall into two categories:

1. **Optimized Initial Images**: These methods employ initial images that closely approximate the input image. For example, the range-averaged image (RAI) replaces range blocks in the initial image with their respective averages:

$$\text{RAI} = \{\text{mean}(R_i) \cdot I\}, \quad i = 1, 2, 3, ..., \text{NumR}$$

Using RAI as the initial image effectively shortens the decoding process [?, ?]. For applications beyond compression, such as fractal image denoising and magnification, the collage image (CI) from the encoding process can further accelerate decoding [?].

2. **Improved Iteration Strategies**: Instead of mapping between two buffers as in conventional methods, one-buffer-decoding (OBD) uses a single buffer, performing mapping operations within the same buffer. This allows previously reconstructed range blocks to assist in reconstructing subsequent ones, accelerating the process [?]. Additionally, a range block

weighted (RBW) method was proposed, where range blocks are weighted by:

$$w_i = \|\alpha_i\| \times \mathrm{count}(R_i)$$

Larger weights indicate easier reconstruction, so these blocks are processed first to assist in reconstructing others, further accelerating decoding [?].

Recently, a fundamentally different MDBS-based method was proposed to reduce decoding complexity [?]. Fractal encoding involves mapping operations from domain blocks to range blocks. However, block matching can be completed using only partial domain blocks, defined as the minimum domain block set (MDBS). Rather than using all domain blocks, only those providing at least one mapping operation for range blocks constitute MDBS:

$$\mathrm{MDBS} = \{D_j \mid \mathrm{Times}(D_j, R_i) \geq 1\}, \quad i = 1, 2, 3, ..., \mathrm{NumR}, \quad j = 1, 2, 3, ..., \mathrm{NumD}$$

where $\mathrm{Times}(A, B)$ denotes the total number of mapping operations from A to B. $\mathrm{NumD}_{\mathrm{MDBS}}$ denotes the total number of domain blocks in MDBS, satisfying $\mathrm{NumD}_{\mathrm{MDBS}} \leq \mathrm{NumD}$. Thus, in each iteration from the first to penultimate, only range blocks within MDBS are reconstructed, saving computations for blocks outside MDBS. In this study, we found redundant domain blocks even within MDBS. By removing these, we obtain OIFS, which further accelerates decoding.

3. Optimal Iterated Function System for Fast Decoding

3.1 Minimum Iterated Function System

We illustrate the minimum iterated function system (MIFS) using Fig. 2 [Figure 2: see original paper]. For a given IFS, Fig. 2 shows mapping operations provided by three domain blocks for all range blocks. In Fig. 2(a), the IFS contains three domain blocks, D_j where $j = 1, 2, 3$, each divisible into four range blocks, R_i where $i = 1, 2, 3, 4$. The symbol "→" represents the mapping operation φ_{ij} from a domain block to a range block. Fig. 2(b) illustrates D_1 and its mappings in red, while Fig. 2(c) shows D_2 and its mappings in green. In Fig. 2(d), D_3 (in blue) provides no mapping operations for any range block. Thus, domain blocks fall into two categories: (1) **Minimum Domain Block Set (MDBS)**: Domain blocks that provide mapping operations for range blocks. Both D_1 and D_2 belong to MDBS. (2) **Redundant Domain Block Set (RDBS)**: Domain blocks that provide no mapping operations. D_3 belongs to RDBS. Removing D_3 and its associated mappings yields the MIFS, defined as:

Definition 1: For an IFS, the domain blocks of MDBS, range blocks within MDBS, and their associated mapping operations constitute the minimum iterated function system (MIFS):

$$\text{MIFS}:\{R_i, D_j, \varphi_{ij}\},\quad i=1,2,3,...,\text{NumR}_{\text{MIFS}},\quad j=1,2,3,...,\text{NumD}_{\text{MIFS}}$$

where $\text{NumR}_{\text{MIFS}}$ and $\text{NumD}_{\text{MIFS}}$ denote the total numbers of range and domain blocks in MIFS, respectively, satisfying $\text{NumD}_{\text{MIFS}} \leq \text{NumD}$. In Fig. 2(e), D_1, D_2, and their mappings constitute the MIFS of the original IFS in Fig. 2(a). Since MIFS domain blocks originate from MDBS, the previous MDBS-based fast decoding method is equivalent to the MIFS-based approach.

3.2 Multilevel Minimum Iterated Function Systems and Optimal Iterated Function System

We illustrate multilevel MIFSs and OIFS using Fig. 3 [Figure 3: see original paper]. Fig. 3(a) shows an IFS with four domain blocks, D_1, D_2, D_3, and D_4, with mappings in red, green, blue, and black, respectively. Here, D_3 provides mappings for some range blocks and belongs to MDBS, while D_4 provides none and belongs to RDBS. Removing D_4 yields the MIFS in Fig. 3(b), identical to the IFS in Fig. 2(a). Treating this MIFS as a new independent IFS reveals that D_3 provides no mappings and belongs to RDBS. Removing D_3 produces the MIFS in Fig. 3(c), identical to Fig. 2(e). When this MIFS is considered independently, no redundant domain blocks remain—all domain blocks (D_1 and D_2) provide mappings for all range blocks within them. Thus, the IFS in Fig. 3(b) is the first-level MIFS of the original IFS, with D_4 as the first-level redundant domain block. The IFS in Fig. 3(c) is the second-level MIFS, with D_3 as the second-level redundant domain block. Finally, the MIFS in Fig. 3(c) contains no redundant domain blocks. Removing D_3 yields the optimal iterated function system (OIFS) in Fig. 3(d), defined as:

Definition 2: For an IFS, OIFS consists of domain blocks that directly or indirectly provide mapping operations for all range blocks without any redundancy, the range blocks within them, and their associated mappings:

$$\text{OIFS}:\{R_i, D_j, \varphi_{ij}\},\quad i=1,2,3,...,\text{NumR}_{\text{OIFS}},\quad j=1,2,3,...,\text{NumD}_{\text{OIFS}}$$

where $\text{NumR}_{\text{OIFS}}$ and $\text{NumD}_{\text{OIFS}}$ denote the total numbers of range and domain blocks in OIFS, satisfying $\text{NumR}_{\text{OIFS}} \leq \text{NumR}$ and $\text{NumD}_{\text{OIFS}} \leq \text{NumD}$. OIFS domain blocks provide mappings for all range blocks within OIFS, enabling independent reconstruction. Moreover, OIFS domain blocks progressively provide mappings for remaining range blocks belonging to different RDBS levels. For example, in Fig. 3, OIFS domain blocks progressively reconstruct range blocks within D_3 and then D_4, ultimately recovering the original IFS.

3.3 Accelerating Fractal Decoding with Optimal Iterated Function System

Table 1 presents the algorithm for calculating OIFS during decoding.

Input: Original IFS
Output: OIFS
Initialization: Let $n = 0$, and $\text{MIFS}^{(0)}$ represents the original IFS. Check each domain block in level-n MIFS, $\text{MIFS}^{(n)}$, to identify level-n RDBS blocks.

while level-n RDBS blocks exist **do** - Remove level-n RDBS blocks - Obtain level-$(n+1)$ MIFS, $\text{MIFS}^{(n+1)}$ - Check each domain block in $\text{MIFS}^{(n+1)}$ for level-$(n+1)$ RDBS blocks

end while

Set $n = N$, and consider level-N MIFS as OIFS.

After encoding, the original IFS is available at decoding. Before the first iteration, OIFS is obtained using Table 1’ s algorithm. The accelerated decoding proceeds as follows: In each iteration from first to penultimate, only range blocks within OIFS are reconstructed, saving computations for blocks outside OIFS. In the final iteration, range blocks within MDBS are reconstructed first based on OIFS, followed by progressive level-by-level reconstruction of blocks outside OIFS (i.e., blocks belonging to different RDBS levels) in reverse order. While the final iteration requires the same computations as previous methods, the overall decoding process is accelerated by saving computations for blocks outside OIFS from the first to penultimate iterations.

4. Experimental Evaluation

4.1 Experimental Settings and Performance Metrics

We evaluated the proposed method using eight 256×256 test images: Airplane, Baboon, Boat, Bridge, Goldhill, House, Lake, and Peppers [Figure 4: see original paper]. Range block size was set to 4×4 with a sliding step of 8. Scaling and offset coefficients s and o were quantized using 5 and 7 bits, respectively. Jacquin’ s method and two state-of-the-art methods (Chaurasia’ s [**?**] and Gupta’ s [**?**]) were used for assessment. Root mean square error (RMSE) measured deviation between iteration and decoded images:

$$\text{RMSE}_n = \sqrt{\frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} [f_n(x,y) - f_{\text{decoded}}(x,y)]^2}, \quad n = 1, 2, ..., N$$

where f_n and f_{decoded} denote iteration and decoded images, respectively, and H and W are image height and width. N represents the total iterations needed for convergence. For iterations 1 through $N-1$, only range blocks within OIFS are reconstructed. The Ratio of $\mathrm{NumD}_{\mathrm{OIFS}}$ to NumD describes the percentage of computations required in each of these iterations:

$$\mathrm{Ratio} = \frac{\mathrm{NumD}_{\mathrm{OIFS}}}{\mathrm{NumD}} \times 100\%$$

If 100% represents computations for reconstructing all range blocks per iteration, the percentage of computations required (PCR) for the entire decoding process is:

$$\mathrm{PCR} = \frac{(N-1) \times \mathrm{Ratio} + 100\%}{N \times 100\%} \times 100\%$$

where numerator and denominator represent computations required by the proposed and previous methods, respectively.

4.2 Implementation Details

The decoded image is obtained by encoding and decoding the input image beforehand. The experimental procedure is:

Step 1: Select a fractal encoding method and encode the input image.
Step 2: Obtain OIFS using Table 1' s algorithm before the first decoding iteration. Initialize $n = 1$.
Step 3: For iteration n, reconstruct all range blocks within OIFS. Calculate RMSE_n for the local image inside OIFS. If convergence requirements are met, proceed to Step 4; otherwise, increment n and repeat Step 3.
Step 4: Reconstruct range blocks belonging to different RDBS levels to obtain the final decoded image.

4.3 Experimental Results

Tables 2-4 show results for each test image. The second and third rows display RMSEs and Ratios for whole iteration images in previous methods. The fourth and fifth rows show those for local iteration images in the MDBS-based method, while the sixth and seventh rows show those for the OIFS-based method. " " indicates convergence. For example, for the Airplane image in Table 2, there are NumD = 1024 domain blocks, and OIFS contains $\mathrm{NumD}_{\mathrm{OIFS}} = 899$ domain blocks (87.79% of the total). In iterations 1 through $N-1$, only OIFS range blocks are reconstructed using OIFS domain blocks with the same operations as previous methods, maintaining identical RMSEs within OIFS. The local image inside OIFS converges when RMSE requirements are satisfied. In the final iteration, range blocks outside OIFS are also reconstructed using OIFS domain blocks. Convergence inside OIFS leads to convergence outside OIFS, and the entire image converges. Thus, the proposed method maintains the same number of iterations as previous methods, as verified by Tables 2-4.

In each of the first to penultimate iterations, the proposed method reconstructs only OIFS range blocks rather than all range blocks, saving 12.21% computations per iteration compared to Jacquin' s method. In the final iteration, both methods reconstruct all blocks. For the Airplane image, $\text{NumD}_{\text{MDBS}} = 942$ (91.99% of total), yielding PCR = 94.66% by Eq. (7), saving 5.34% computations for MDBS versus Jacquin' s method. With $\text{NumD}_{\text{OIFS}} = 899$ (87.79%), PCR = 91.86%, saving an additional 2.80% for OIFS versus MDBS.

For Jacquin' s method on all eight test images, MDBS-based decoding required 91.99%, 85.35%, 89.65%, 87.21%, 90.14%, 88.48%, 90.04%, and 92.38% computations per iteration (iterations 1 to $N-1$). The corresponding PCR values were 94.66%, 90.23%, 92.24%, 89.34%, 93.43%, 91.36%, 92.53%, and 94.92%. OIFS-based decoding required 87.79%, 77.54%, 82.91%, 81.35%, 87.50%, 85.64%, 87.30%, and 90.82% computations per iteration, with PCR values of 91.86%, 85.03%, 87.18%, 84.46%, 91.67%, 89.23%, 90.47%, and 93.88%. Thus, OIFS saved an additional 2.80%, 5.20%, 5.06%, 4.88%, 1.76%, 2.13%, 2.06%, and 1.04% compared to MDBS.

For Chaurasia' s method, MDBS-based decoding required 81.05%, 81.15%, 81.25%, 78.81%, 81.45%, 76.86%, 79.59%, and 83.01% computations per iteration, with PCR values of 87.37%, 87.43%, 87.50%, 84.11%, 87.63%, 82.64%, 84.69%, and 88.67%. OIFS-based decoding required 73.14%, 68.55%, 70.31%, 64.55%, 68.85%, 70.21%, 70.31%, and 74.32% computations per iteration, with PCR values of 82.09%, 79.03%, 80.21%, 73.41%, 79.23%, 77.66%, 77.73%, and 82.88%. OIFS saved 5.28%, 8.40%, 7.29%, 10.70%, 8.40%, 4.98%, 6.96%, and 5.79% compared to MDBS.

For Gupta' s method, MDBS-based decoding required 84.18%, 81.64%, 83.01%, 80.76%, 83.69%, 80.86%, 83.98%, and 84.18% computations per iteration, with PCR values of 89.45%, 87.76%, 87.26%, 85.57%, 89.13%, 85.64%, 87.99%, and 89.45%. OIFS-based decoding required 78.81%, 72.17%, 76.17%, 76.63%, 78.42%, 76.27%, 78.32%, and 78.91% computations per iteration, with PCR values of 85.87%, 81.45%, 82.13%, 80.22%, 85.61%, 82.20%, 83.74%, and 85.94%. OIFS saved 3.58%, 6.31%, 5.13%, 5.35%, 3.52%, 3.44%, 4.25%, and 3.51% compared to MDBS.

Overall, the OIFS-based method saves approximately 1%-10% of total computations compared to the MDBS-based method.

4.4 Analysis of Experimental Results

For Jacquin' s method, Table 5 shows domain block counts across MIFS levels for all test images. Using the Airplane image as an example, there are 1024 total domain blocks. The first through fourth-level MIFSs contain 942, 914, 901, and 899 domain blocks, respectively. Compared to the original method, MDBS saves computations for 1024 - 942 = 82 domain blocks. The OIFS-based method saves an additional 942 - 899 = 43 domain blocks. Similar savings are achieved for other images: 80, 69, 60, 27, 29, 28, and 16 domain blocks.

Fig. 5 [Figure 5: see original paper] illustrates domain blocks for the eight test images. White and green blocks together represent MDBS domain blocks, while white blocks alone represent OIFS domain blocks. Thus, green blocks represent domain blocks from second to last RDBS levels, whose computations are saved compared to the MDBS-based method.

For Chaurasia' s method, Table 6 shows MIFS-level domain block counts. The MDBS-based method reconstructs 830, 831, 832, 807, 834, 787, 815, and 850 domain blocks per iteration (iterations 1 to $N-1$), while the OIFS-based method reconstructs 749, 702, 720, 661, 706, 719, 720, and 761 domain blocks. This saves 81, 129, 112, 41, 128, 68, 95, and 89 domain blocks for OIFS versus MDBS. In Fig. 6 [Figure 6: see original paper], white and green blocks represent MDBS domain blocks, with white blocks alone representing OIFS domain blocks. Green blocks thus represent second to last RDBS-level domain blocks.

For Gupta' s method, Table 7 shows domain block counts. The MDBS-based method reconstructs 862, 836, 850, 827, 857, 828, 860, and 862 domain blocks per iteration, while OIFS reconstructs 807, 739, 780, 754, 803, 781, 802, and 808 domain blocks, saving 55, 97, 70, 73, 54, 47, 58, and 54 domain blocks. Fig. 7 [Figure 7: see original paper] similarly illustrates MDBS (white and green) and OIFS (white only) domain blocks.

5. Conclusion

This paper proposes an OIFS-based fast fractal decoding method. We introduced definitions for MIFS, multilevel MIFS, and OIFS. During decoding, OIFS achieves independent convergence within OIFS using the same number of iterations as conventional methods, saving computations for reconstructing range blocks outside OIFS. However, the final iteration must reconstruct all range blocks (both inside and outside OIFS) to recover the complete decoded image. We validated the proposed method using Jacquin's method and two state-of-the-art approaches. By reducing decoding complexity, the proposed method makes fractal image coding more suitable for real-time applications such as remote monitoring, video conferencing, and virtual reality.

Conflict of Interests: The author declares no conflict of interests regarding this paper' s publication.

References

[1] Barnsley, M. *Fractals Everywhere*. Academic Press, New York: 1988.
[2] Jacquin AE. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans Image Process* 1992; 1(1): 18-30. https://doi.org/10.1109/83.128028.
[3] Fisher Y. *Fractal Image Compression: Theory and Application*. Springer-Verlag; 1994.
[4] Wohlberg B, Jager G de. A review of the fractal image coding literature.

IEEE Trans Image Process 1999; 8(12): 1716-1729. https://doi.org/10.1109/83.806618.
[5] Jacquin AE. Fractal image coding: A review. *Proc IEEE* 1993; 81(10): 1451-1465. https://doi.org/10.1109/5.241507.
[6] Chaurasia V, Chaurasia V. Statistical feature extraction based technique for fast fractal image compression. *J Vis Commun Image R* 2016; 41: 87-95. https://doi.org/10.1016/j.jvcir.2016.09.008.
[7] Gupta R, Mehrotra D, Tyagi RK. Hybrid edge-based fractal image encoding using K-NN search. *Multimed Tools Appl* 2022; 81(15): 21135-21154. https://doi.org/10.1007/s11042-022-12631-7.
[8] Shen FR, Osamu H. A fast no search fractal image coding method. *Signal Process Image Commun* 2004; 19(5): 393-404. https://doi.org/10.1016/j.image.2004.02.002.
[9] Wang XY, Wang SG. An improved no-search fractal image coding method based on a modified gray-level transform. *Comput Graph* 2008; 32(4): 445-450. https://doi.org/10.1016/j.cag.2008.02.004.
[10] Wang XY, Wang YX, Yun JJ. An improved no-search fractal image coding method based on a fitting plane. *Image Vision Comput* 2010; 28, 1303-1308. https://doi.org/10.1016/j.imavis.2010.01.008.
[11] Bi S, Wang Q. Fractal image coding based on a fitting surface. *J Appl Math* 2014; 634848. https://doi.org/10.1155/2014/634848.
[12] Nandi U. Fractal image compression using a fast affine transform and hierarchical classification scheme. *The Visual Computer* 2022; 38: 3867–3880. https://doi.org/10.1007/s00371-021-02226-y.
[13] Roy SK, Kumar S, Chanda B, Chaudhuri BB, Banerjee S. Fractal image compression using upper bound on scaling parameter. *Chaos Solitons & Fractals* 2018; 106: 16-22. https://doi.org/10.1016/j.chaos.2017.11.013.
[14] Nandi U. Fractal image compression with adaptive quadtree partitioning and non-linear affine map. *Multimed Tools Appl* 2020; 79: 26345–26368. https://doi.org/10.1007/s11042-020-09256-z.
[15] Nandi U. An adaptive fractal-based image coding with hierarchical classification strategy and its modifications. *Innovations Syst Softw Eng* 2019; 15: 35–42. https://doi.org/10.1007/s11334-019-00327-5.
[16] Wang Q, Jin GH, Bi S. Adaptively hybrid fractal image coding. *IET Image Process* 2024; 18(7): 1745-1758. https://doi.org/10.1049/ipr2.13060.
[17] Pi MH, Mandal MK, Basu A. Image retrieval based on histogram of fractal parameters. *IEEE Trans Multimedia* 2005; 7(4): 597-605. https://doi.org/10.1109/TMM.2005.846796.
[18] Wang XY, Chen Z. A fast fractal coding in application of image retrieval. *Fractals* 2009; 17(4): 441-450. https://doi.org/10.1142/s0218348x09004557.
[19] Huang X, Zhang Q, Liu W. A new method for image retrieval based on analyzing fractal coding characters. *J Vis Commun Image R* 2013; 24(1): 42-47. https://doi.org/10.1016/j.jvcir.2012.10.005.
[20] Pi MH, Li CH, Li H. A novel fractal image watermarking. *IEEE Trans Multimedia* 2006; 8(3): 488-499. https://doi.org/10.1109/tmm.2006.870738.
[21] Daraee F, Mozaffari S. Watermarking in binary document images using fractal codes. *Pattern Recognit Lett* 2014; 35: 120-129. https://doi.org/10.1016/j.patrec.2013.04.022.

[22] Lu J, Zou YR, Yang CY, Wang LJ. A robust fractal color image watermarking algorithm. *Math Probl Eng* 2014: 638174. https://doi.org/10.1155/2014/638174.
[23] Chung KH, Fung YH, Chan YH. Image enlargement using fractal. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* 2003; VI-273: 273-275. https://doi.org/10.1109/icassp.2003.1201671.
[24] Lai CM, Lam KM, Chan YH, Siu WC. An efficient fractal based algorithm for image magnification. In: *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, 2004; ii-iii: 571-574. https://doi.org/10.1109/ISIMP.2004.1433972.
[25] Chen ZP, Ye ZL, Wang SX, Peng GH. Image magnification based on similarity analogy. *Chaos Solitons & Fractals* 2009; 40(5): 2370-2375. https://doi.org/10.1016/j.chaos.2007.10.031.
[26] Wee YC, Shin HJ. A novel fast fractal super resolution technique. *IEEE Trans Consumer Electron* 2010; 56(3): 1537-1541. https://doi.org/10.1109/tce.2010.5606294.
[27] Hua Z, Zhang HC, Li JJ. Image super resolution using fractal coding and residual network. *Complexity* 2019; 9419107. https://doi.org/10.1155/2019/9419107.
[28] Liu S, Bai WL, Zeng NY, Wang SH. A fast fractal based compression for MRI images. *IEEE Access* 2019; 7: 62412-62420. https://doi.org/10.1109/access.2019.2916934.
[29] Abdullahi SM, Wang HX, Li T. Fractal coding-based robust and alignment-free fingerprint image hashing. *IEEE Trans Inform Forensic Secur* 2020; 15: 2587-2601. https://doi.org/10.1109/tifs.2020.2971142.
[30] Khelaifi F, He HJ. Perceptual image hashing based on structural fractal features of image coding and ring partition. *Multimed Tools Appl* 2020; 79(27-28): 19025-19044. https://doi.org/10.1007/s11042-020-08619-w.
[31] Bisogni C, Nappi M, Pero C, Ricciardi S. FASHE: A fractal based strategy for head pose estimation. *IEEE Trans on Image Process* 2021; 30: 3192-3203. https://doi.org/10.1109/TIP.2021.3059409.
[32] Bisogni C, Nappi M, Pero C, Ricciardi S. PIFS scheme for head pose estimation aimed at faster face recognition. *IEEE Trans on Biometrics, Behavior, and Identity Science* 2022; 4: 173-184. https://doi.org/10.1109/TBIOM.2021.3122307.
[33] Ghazel M, Freeman GH, Vrscay ER. Fractal image denoising. *IEEE Trans Image Process* 2003; 12(12): 1560-1578. https://doi.org/10.1109/TIP.2003.818038
[34] Ghazel M, Freeman GH, Vrscay ER. Fractal-wavelet image denoising revisited. *IEEE Trans Image Process* 2006; 15(9): 2669-2675. https://doi.org/10.1109/tip.2006.877377.
[35] Lu J, Ye ZX, Zou YR, Ye RS. An enhanced fractal image denoising algorithm. *Chaos Solitons & Fractals* 2008; 38(4): 1054-1064. https://doi.org/10.1016/j.chaos.2007.06.048.
[36] Jeng JH, Tseng CC, Hsieh JG. Study on huber fractal image compression. *IEEE Trans Image Process* 2009; 18(5): 995-1003. https://doi.org/10.1109/tip.2009.2013080.
[37] Lu J, Ye ZX, Zou YY. Huber fractal image coding based on a fitting plane. *IEEE Trans Image Process* 2013; 22(1): 134-145. https://doi.org/10.1109/tip.2012.2215619.
[38] Zou YY, Hu HX, Lu J, Liu XX, Jiang QT, Song GH. A nonlocal low-rank regularization method for fractal image coding. *Fractals* 2021; 29(5): 2150125. https://doi.org/10.1142/s0218348x21501255.
[39] Xu C, Ye YT, Hu ZW, Zou YR, Shen LX, Liu XX, Liu J. A primal-dual

algorithm for robust fractal image coding. *Fractals* 2019; 27(7): 1950119. https://doi.org/10.1142/s0218348x19501196.
[40] Pan H, Liang ZY, Lu J, Tu K, Xie N. Nonlocal low rank regularization method for fractal image coding under salt-and pepper noise. *Fractals* 2023; 31(7): 2350076. https://doi.org/10.1142/S0218348X23500767.
[41] Liang WJ, Li XY, Tu ZH, Lu J. Multiparent fractal image coding-based methods for salt-and-pepper noise removal. *Fractals* 2024; 32(1): 2450012. https://doi.org/10.1142/S0218348X24500129.
[42] Wang J., Zhang M., Tong X., Wang Z.: A chaos-based image compression and encryption scheme using fractal coding and adaptive-thresholding sparsification. *Phys. Scr.* 97(2022) 10521. https://doi.org/10.1088/1402-4896/ac8b41.
[43] Liu, X., Tong, X., Zhang, M. et al. Image compression and encryption algorithm based on uniform non-degeneracy chaotic system and fractal coding. *Nonlinear Dyn.* 111, 8771-8798 (2023). https://dio.org/10.1007/s11071-023-08281-5.
[44] Long, B., Chen, Z., Liu T. et al. Improved fractal coding and hyperchaotic system for lossless image compression and encryption. *Nonlinear Dyn.* (2024). https://doi.org/10.1007/s11071-024-10671-2.
[45] Moon Y.H., Kim, J.S., Kim, J.H.: a fast fractal decoding algorithm based on the selection of an initial image. *IEEE Trans Image Process.* 5 (2000), 941-945. https://doi.org/10.1109/83.841539.
[46] Huang X.Y., He C.J., Huang H.M.: Improved scheme for fast fractal decoding algorithm based on initial image selection. *Dynamics of Continuous Discrete and Impulsive Systems-Series B-Applications & Algorithms.* 2 (2005), 625-629.
[47] Wang Q., Bi S., Liang D.Q.: A novel selection of the initial image for fast fractal decoding. *ICIC Express Lett. Part B Appl.* 4 (2013), no.5, 1253-1258
[48] Kang, H.S., Kim, S.D.: Fractal decoding algorithm for fast convergence. *Opt. Eng.* 35(1996), no.11, 3191-3198. https://doi.org/10.1117/1.601058.
[49] Wang Q.: An improved fast fractal decoding method. *ICIC Express Lett. Part B Appl.* 7 (2016), no.9, 1951-1956. https://doi.org/10.2507/icicelb.07.09.1951.
[50] Wang Q.: Accelerated decoding method in fractal image coding. *Electron. Lett.* 60 (2024), no.24, 1-5. https://doi.org/10.1049/ell2.70120.

Note: Figure translations are in progress. See original paper for figures.