Fast Fractal Image Decoding Based on Minimum Iterated Function System

Authors: Wang, Qiang, Wang, Qiang

Date: 2023-09-19T00:00:00+00:00

Abstract

To accelerate the fractal decoding process, a minimum iterated function system based fast fractal decoding method was proposed in this study. In fractal encoding process, we found that there exists a minimum domain block set (MDBS) which can provide the best-matched domain blocks for all range blocks, and then the domain blocks of MDBS, the range blocks within MDBS, and the associated mapping operations between them constitute the minimum iteration function system (MIFS). In decoding process, MIFS was first recovered in the first iteration. Then, in each of the second to penultimate iterations, only the range blocks within MDBS are reconstructed, and the computations of reconstructing the remaining range blocks can be saved. Finally, all range blocks are reconstructed to obtain the decoded image in the last iteration. Four fractal encoding methods were adopted to assess the performance of the proposed method. Experimental results show that the proposed method can complete the decoding process with fewer computations.

Full Text

Preamble

Fast Fractal Image Decoding Based on Minimum Iterated Function System

Wang Qiang*
College of Information Science and Technology, Dalian Maritime University
Linghai Road, No.1, Dalian, China, 116026
*Corresponding Author, E-mail: wangqiang2011@dlmu.edu.cn

Abstract: To accelerate the fractal decoding process, this study proposes a fast fractal decoding method based on a minimum iterated function system. In the fractal encoding process, we discovered that a minimum domain block set (MDBS) exists that can provide the best-matched domain blocks for all range blocks. The domain blocks within the MDBS, the range blocks contained in the MDBS, and the associated mapping operations between them constitute the minimum iterated function system (MIFS). During decoding, the MIFS is first recovered in the initial iteration. Then, in each subsequent iteration from the second to the penultimate, only the range blocks within the MDBS are reconstructed, thereby saving the computational cost of reconstructing the remaining range blocks. Finally, all range blocks are reconstructed in the last iteration to obtain the decoded image. Four fractal encoding methods were employed to evaluate the performance of the proposed method. Experimental results demonstrate that the proposed method can complete the decoding process with significantly fewer computations.

Keywords: Fractal image decoding; Minimum iterated function system; Minimum domain block set; Redundant domain block set; Fixed point

1. Introduction

The foundational concept of fractal image coding was first introduced by Barnsley, and subsequently, Jacquin proposed the first practical fractal coding algorithm, which was regarded as a promising approach for image compression [1,2]. Despite suffering from high computational complexity during the encoding process, fractal image coding offers several distinctive advantages, including a novel theoretical foundation, potential for high compression ratios, fast decoding, and resolution independence. Consequently, researchers worldwide have developed various fast fractal encoding methods, which can be broadly categorized into two approaches. The first category comprises local block matching-based fast fractal image coding methods [3-5], which accelerate the encoding process by converting exhaustive block matching into localized search within the domain block pool while striving to maintain decoded image quality. The second category includes no-search fractal image coding methods [6-9], which directly assign the best-matched domain block without performing block matching operations, thereby enabling real-time encoding at the cost of degraded decoded image quality. Meanwhile, after years of research, fractal image coding has found applications in numerous other image processing domains, including image denoising [10-16], image magnification [17-21], image hashing [22,23], image retrieval [24-26], watermarking [27-29], and human pose estimation [30,31].

In this study, we propose a novel method to accelerate the fractal decoding process. The fractal encoding process typically consists of a series of mapping operations from domain blocks to range blocks. Our investigation revealed that block matching operations can be completed using only a subset of domain blocks, which we define as the minimum domain block set (MDBS). Consequently, the remaining domain blocks outside the MDBS are designated as the redundant domain block set (RDBS). Furthermore, the domain blocks within the MDBS, the range blocks contained in the MDBS, and their associated mapping operations collectively constitute the minimum iterated function system (MIFS). Based on these definitions, it is evident that by excluding the domain blocks from the RDBS, the MIFS contains the essential mapping operations from the MDBS to all range blocks and plays a crucial role in the encoding process. During decoding, all range blocks are reconstructed in the first iteration, enabling the recovery of the MIFS by distinguishing the MDBS from the RDBS. In each subsequent iteration from the second to the penultimate, only the range blocks within the MDBS are reconstructed using the MIFS. In the final iteration, the local image within the MDBS converges, and the range blocks within the RDBS are subsequently reconstructed. In summary, the proposed method performs identical operations to conventional methods in the first and last iterations. However, for iterations two through N-1, since the domain blocks within the RDBS provide no mapping operations for any range block, there is no need to reconstruct the range blocks within the RDBS, thereby saving the associated computational cost. We evaluated the performance of our proposed method using three state-of-the-art methods alongside Jacquin' s original method. Experimental results demonstrate that compared with conventional methods, the proposed approach effectively reduces the number of computations required in the decoding process.

2. Conventional Fractal Image Coding

Conventional fractal image encoding, as proposed by Jacquin, can be described in detail as follows [2]. The M × N input image is first partitioned into non-overlapping B × B range blocks, denoted as i i =R 1, 2,3,⋯, NumR , where NumR represents the total number of range blocks. Domain blocks are then obtained by sliding a 2B × 2B window over the input image with a fixed sliding step δ=2B, yielding 1, 2,3,⋯, NumD domain blocks, where NumD denotes the total number of domain blocks. All domain blocks are subsequently contracted to size B × B and constitute a domain block pool, which is extended with eight isometric transformations. Finally, for any given range block Ri, the best-matched domain block Dj can be found by minimizing the following function: = argmin = argmin 1, 2,3,⋯, NumR, 1 , 2,3,⋯, NumD where ijϕ ⋅denotes the mapping operation from jD to ( ) ⋅( )γ ⋅represent the contracting and isometric transformations, respectively, and iα and iβ denote the scaling and offset coefficients of the affine transformation, respectively. I denotes a B × B block whose components are all ones.

In the decoding process, any M × N image can be selected as the initial image. In each subsequent iteration, all range blocks are reconstructed by their respective best-matched domain blocks based on the same operations used in the encoding process. After approximately ten iterations, the decoding process converges to the final decoded image. When the Bridge and Lake images are used as the input image for encoding and the initial image for decoding, respectively, Figure 1(a) illustrates the initial image, while the remaining five images in Figures 1(b)-(f) show the first five iteration images during the decoding process. We observe that the initial image gradually converges to the input image.

[Figure 1: see original paper]

3. Accelerating Fractal Decoding Process by Minimum Iterated Function System

In the encoding process, range blocks can be approximated by their respective best-matched domain blocks through a series of contracting, isometric, and affine transformations as follows: 1, 2,3,···, NumR, 1, 2,3,···, NumD where j D denotes the best-matched domain block of iR , NumD denotes the total number of best-matched domain blocks actually required, and satisfies NumD NumD . Although all domain blocks 1, 2,3,···, NumD are candidates for the best-matched domain block of iR , only partial domain blocks 1, 2,3,···, NumD are actually needed. The remaining domain blocks cannot provide mapping operations for any range block and are never utilized. For example, in Figure 2(a), we have three domain blocks, 1, 2,3 , each of which can be uniformly divided into four range blocks, ji i = 1, 2,3, 4 . The arrow "→" represents the mapping operation ijϕ ・from one domain block to another range block. Figures 2(b), (c), and (d) illustrate the mapping operations provided by domain blocks 2D and 3D , respectively. We observe that in Figures 2(b) and (c), both 1D and 2D can provide mapping operations for range blocks. However, in Figure 2(d), 3D provides no mapping operation for any range block. Thus, we can categorize domain blocks into two sets: the minimum domain block set (MDBS) and the redundant domain block set (RDBS).

Definition 1: The domain blocks that can provide mapping operations for range blocks at least once constitute the minimum domain block set (MDBS): s.t. Times 1, 2,3,···, NumR, 1, 2,3,···, NumD where Times(A, B) denotes the total number of mapping operations from A to B. From Definition 1, we know that the domain blocks of the MDBS can provide mapping operations for all range blocks (including those within the MDBS). In Figure 2, both 1D and represent the domain blocks of the MDBS. We then define the redundant domain block set (RDBS) as follows.

Definition 2: The domain blocks that provide no mapping operation for any range block constitute the redundant domain block set (RDBS): s.t. Times 1, 2,3,···, NumR, 1, 2,3,. .., NumD where NumD denotes the total number of domain blocks in the RDBS and satisfies NumD =NumD NumD . In Figure 2, 3D represents a domain block of the RDBS. For the Lake and Bridge images shown in Figures 3(a) and (b), the domain blocks within the white and red boxes represent the domain blocks of the MDBS and RDBS, respectively.

[Figure 2: see original paper]

[Figure 3: see original paper]

We further define the minimum iterated function system (MIFS) as follows.

Definition 3: The domain blocks of the MDBS, the range blocks within the MDBS, and the associated mapping operations between them are defined as the minimum iterated function system (MIFS):

MIFS : 1, 2,3,···, NumR , 1, 2,3,···, NumD , MDB S where 1, 2,3,···, NumR denote the range blocks within the MDBS. NumR denotes the total number of range blocks within the MDBS and satisfies NumR NumD 4 . From Definitions 1, 2, and 3, we know that because the MIFS can provide mapping operations from the domain blocks of the MDBS to the range blocks within the MDBS, the MIFS can independently complete the mapping operations by itself. Moreover, because the domain blocks of the MDBS within the MIFS can also provide mapping operations for the range blocks within the RDBS, the MIFS can be used to reconstruct the entire decoded image during the decoding process. Figure 4 illustrates the MIFS contained in Figure 2(a).

[Figure 4: see original paper]

The proposed decoding method can be designed as follows. In the first iteration, based on the mapping operations from the encoding process, we reconstruct all range blocks, and the MIFS can be recovered by distinguishing the MDBS from the RDBS. Then, in each iteration from the second to the penultimate, only the range blocks within the MDBS are reconstructed by the domain blocks of the MDBS based on the MIFS. On one hand, for the range blocks within the MDBS, the MIFS maintains the same reconstruction operations as conventional methods. Thus, we obtain the same local iteration image within the MDBS as before, which can be used to determine whether convergence has been achieved, and the same number of iterations can be maintained. On the other hand, based on Definition 2, because the domain blocks of the RDBS do not participate in reconstructing any range block, there is no need to reconstruct the range blocks within the RDBS, and the associated computations can be saved. Finally, in the last iteration, after reconstructing the range blocks within the MDBS, the range blocks within the RDBS are also reconstructed, and the complete decoded image can be obtained. Figure 5 illustrates the process of the proposed decoding method, showing that it maintains the same number of iterations as conventional methods. In the first and last iterations, the unshaded regions represent all range blocks within both MDBSs and RDBSs that need to be reconstructed, and the proposed method performs the same operations as conventional methods. For each iteration from the second to the penultimate, the unshaded regions still represent the range blocks within MDBSs that need to be reconstructed. However, the shaded regions represent the range blocks within RDBSs whose reconstruction operations can be saved, thereby accelerating the decoding process.

[Figure 5: see original paper]

4. Experiments

In this section, two 256 × 256 images, Lake and Bridge, are selected as test images. The range block size is set to 4 × 4, and the sliding step is 8. The scaling and offset coefficients, s and o, are quantized with 5 and 7 bits, respectively. Jacquin’ s method and three state-of-the-art methods—Jacquin’ s [2], Chaurasia' s [3], Zheng' s [4], and Gupta' s methods [5]—are adopted to evaluate the performance of the proposed method. The root mean square error (RMSE) is used to measure the deviation between the kth iteration image and the decoded image as: × f Decoded =1, 2,···, where f(k) and fDecoded denote the kth iteration and decoded images, respectively, H and W represent the image height and width, respectively, and N denotes the total number of iterations required to complete the decoding process. In the experiments, the decoded image is obtained by encoding and decoding the input image in advance, and the detailed experimental procedures are as follows:

Step 1: Select a fractal encoding method and encode the input image.

Step 2: In the first iteration of the decoding process, select a 256 × 256 blank image as the initial image, reconstruct all range blocks, and partition all domain blocks into MDBS and RDBS. Set k=1, calculate and record RMSE(k) within the MDBS.

Step 3: Reconstruct the range blocks within the MDBS. Set k=k+1, calculate and record RMSE(k) within the MDBS. If the convergence requirement of the local image within the MDBS is satisfied (≤), proceed to Step 4. Otherwise, return to Step 3 and perform another iteration.

Step 4: Reconstruct the range blocks within the RDBS to obtain the final decoded image.

In Tables 1, 2, 3, and 4, " " indicates convergence achieved in the Nth iteration. For the Lake image in Table 1, we have NumD=1024 domain blocks total, with the MDBS and RDBS containing NumD =922 and Num =102 domain blocks, respectively, comprising 90.04% and 9.96% of the input image. In each iteration from the second to the (N-1)th, only the range blocks within the MDBS are reconstructed. On one hand, these range blocks are reconstructed by the domain blocks of the MDBS, implying that the same reconstruction operations are performed within the MDBS as in Jacquin' s method. Thus, the proposed method maintains the same RMSE values within the MDBS as Jacquin's method and preserves the same number of iterations. When the deviation between the current and previous local iteration images within the MDBS satisfies the ≤ requirement, the decoding process is considered to have converged. On the other hand, because the proposed method reconstructs only the range blocks within the MDBS rather than all range blocks as in Jacquin' s method, 9.96% of the total computations can be saved compared to Jacquin' s method. Finally, all range blocks within both MDBS and RDBS are reconstructed in the last iteration. For the entire decoding process, the proposed method performs identically to Jacquin' s method in the first and last iterations, while reconstructing only the range blocks within the MDBS for iterations two through N-1. If the percentage of computations required (PCR) for reconstructing all range blocks in each iteration is considered 100%, the PCR for the proposed method across the entire decoding process can be calculated as: 2) Num NumD 100% + 2 100% where the numerator and denominator represent the computations required by the proposed and conventional methods, respectively. Based on Eq. (7), for the Lake image in Table 1 , we have Num NumD 100%=90.04% and N=11, yielding PCR=91.85%, which means 8.15% of total computations are saved compared to Jacquin' s method. Similarly, for the Bridge image, the MDBS and RDBS contain NumD =893 and Num =131 domain blocks, respectively, with N=11. Using Eq. (7), we obtain PCR=89.54%, saving 10.46% of total computations.

Tables 2, 3, and 4 present the experimental results for Chaurasia' s, Zheng' s, and Gupta' s methods, respectively. In Table 2 , for Lake and Bridge images, 81.35% and 79.98% of total computations are required in each iteration from the second to ninth, with PCR values of 84.74% and 83.62%, respectively. In Table 3 , for both test images, 90.14% and 87.21% of total computations are required in each iteration from the second to tenth, with PCR values of 91.93% and 89.54%, respectively. In Table 4 , for both test images, 90.72% and 88.09% of total computations are required in each iteration from the second to tenth, with PCR values of 92.41% and 90.26%, respectively. In summary, compared with conventional methods, the proposed method effectively reduces the number of computations required in the decoding process.

5. Conclusion

This paper proposes a novel fast fractal decoding method. In the encoding process, we introduce the definitions of MDBS, RDBS, and MIFS. During decoding, the MIFS can complete the decoding process and achieve convergence within the MDBS independently using the same number of iterations as conventional methods, thereby saving the computations required to reconstruct the range blocks within the RDBS from the second to penultimate iterations. The reconstruction of range blocks within the RDBS in the first and last iterations must still be performed to recover the MIFS and obtain the complete decoded image, respectively. Finally, Jacquin' s method and three state-of-the-art methods are employed to verify the effectiveness of the proposed approach.

References

1. B. Wohlberg and B. G. Jager, "A review of the fractal image coding literature," IEEE Trans. on Image Process. 8(12), 1716-1729 (1999).

2. A. K. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations,"IEEE Trans. on Image Process. 1(1), 18-30 (1992)

3. V. Chaurasia and V. Chaurasia, "Statistical feature extraction based technique for fast fractal image compression," J. Vis. Commun. Image R. 41, 87-95 (2016).

4. Y. P. Zheng, X. P. Li, and M. Sarem, "Fast fractal image compression algorithm using specific update search," IET Image Process. 14(9), 1733-1739 (2020).

5. R. Gupta, D. Mehrotra, and R. K. Tyagi, "Hybrid edge-based fractal image encoding using K-NN search," Multimed. Tools Appl. 81, 21135-21154 (2022).

6. F. R. Shen and H. Osamu, "A fast no search fractal image coding method," Signal Process. Image Commun. 19(5), 393-404 (2004).

7. X. Y. Wang and S. G. Wang, "An improved no-search fractal image coding method based on a modified gray-level transform," Comput. Graph. 32(4), 445-450 (2008).

8. X. Y. Wang, Y. X. Wang, and J. J. Yun, "An improved no-search fractal image coding method based on a fitting plane," Image Vision Comput. 28, 1303-1308 (2010).

9. S. Bi and Q. Wang, "Fractal image coding based on a fitting surface" J. Appl. Math. 2014, 634848 (2014).

10. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," IEEE Trans. on Image Process. 12(12), 1560-1578 (2003).

11. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," IEEE Trans. on Image Process. 15(9), 2669-2675 (2006).

12. J. Lu, Z. X. Ye, Y. Y. Zou, and R. S. Ye, "An enhanced fractal image denoising algorithm," Chaos, Soliton. Fract. 38, 1054-1064 (2008).

13. J. H. Jeng, C. C. Tseng, and J. G. Hsieh, "Study on huber fractal image compression," IEEE Trans. on Image Process. 18(5), 995-1003 (2009).

14. J. Lu, Z. X. Ye, and Y. Y. Zou, "Huber fractal image coding based on a fitting plane," IEEE Trans. on Image Process. 22(1), 134-145 (2013).