A simplified fractal image compression algorithm

Conference Paper · December 2008

DOI: 10.1109/ICCES.2008.4772965 · Source: IEEE Xplore

Mohiy M. Hadhoud
Minoufiya University

M.I. Dessouky
Minoufiya University

Fathi E. Abd El-Samie
Minoufiya University

A Simplified Fractal Image Compression Algorithm

A. Selim*, M. M. Hadhoud, M. I. Dessouky*** and F. E. Abd El-Samie*****

*ERTU,Egypt

** Dept. of Information Technology, Faculty of Computers and Information , Menoufia University, 32511, Shebin Elkom , Egypt.

*** Dept. of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University, 32952, Menouf , Egypt.

E-mails: a_seleem_87@yahoo.com , mmhadhoud@yahoo.com, dr_moawad@yahoo.com and fathi_sayed@yahoo.com

***ABSTRACT*- This paper proposes a simplified fractal image compression algorithm which is implemented on a block by block basis. This algorithm achieves a compression ratio of up to 10 with a peak signal to noise ratio (PSNR) as high as 35dB. The idea of the proposed algorithm is based on the segmentation of the image, first, into blocks to setup reference blocks. The image is then decomposed again into block ranges and a search process is carried out to find the reference blocks with best match. The transmitted or stored values, after compression, are the reference block values and the indices of the reference block that achieves the best match. If there is no match, the average value of the block range is transmitted or stored instead. The advantages of the proposed algorithm are the simplicity of computation and the high PSNR achieved.**

I. INTRODUCTION

There are many techniques used to compress both image and video. One of them is the fractal which is based on partitioning of an image into regions or blocks. It doesn't depend on the translation of the image to the frequency domain as in the case of differential pulse code modulation (DPCM), which depends on the discrete cosine, transform (DCT). DPCM suffers from blocking artifacts which is not the case in fractal image compression (FIC). Fractal image compression is one of the most popular modern image compression methods. It uses the self similarly concept and has many features such as, long coding time, fast decoding, high compression ratio, and independence on resolution [1-3]. Barnsly was the first one who gave the attractor model of the two dimension affine transform. He developed methods to compress an image based on the iterated function system (IFS) [8-10].Generally FIC is a lossy compression technique.

In fractal image compression, the image is divided into a number of block domains with arbitrary size (ranging from 2x2 to 16**x**16, or more). Then, the image is divided again into block ranges with size less than that of the block domain. There are several methods that can be adopted to obtain the attractor model necessary to decode an image. Examples of these methods are block segmentation, region segmentation and the cross searching method [1, 2]. All these methods lie under the umbrella of self similarity algorithms. All these methods also imply exhaustive computation algorithms. Sections (3.1) and (3.2) are devoted to the explanation of block segmentation and region segmentation basics.

In our proposed algorithm, the image is divided into equal square regions, and then a search process is carried out to find the reference block in each region. The selected reference blocks are used to formulate the reference block domain pool. The image is divided again into block ranges. Then a search is performed in the reference block domain pool for the best match with each range block. The only transmitted or stored data are the indices of the selected reference block for each range block, instead of the range itself. If there is no matched reference block, according to a certain threshold, the average value of the range block is transmitted instead of the block itself. In this algorithm, we don't use transformations which increase the complexity of the encoder. We use the absolute difference to determine the similarity between blocks.

The resulting images of the proposed algorithm have high PSNR with moderate compression ratio. The main advantage of the proposed algorithm is the simplicity in encoding and decoding.

The paper is organized as follows: section II gives the basics of fractal image compression. Section III surveys two types of segmentation in fractal image compression. In section IV, the proposed simplified fractal compression algorithm is presented. In section V, the experimental results are presented. Finally the concluding results are given in section VI.

II. BASICS OF FRACTAL IMAGE COMPRESSION

The word fractal comes from the Latin word fractus, meaning broken. The word describes objects that are irregular to fit into traditional geometrical shapes. There are several definitions that have been proposed for this word. One of them defines a fractal as a set with Hausdorff dimension strictly greater than its Euclidean dimension. Thus, a fractal has a factional dimension. This has lead to the thought that the word fractal has come from the fractional dimension. The most popular definition of fractal claims that it is a geometric form whose irregular details recur at different scales and angles which can be described by affine or fractal transforms [7-13].

Strictly speaking, it can be said that a fractal has the following properties:

1. It has a fine structure, i.e., details on arbitrarily small scales.
2. It is too irregular to be described in a traditional geometrical language, both locally and globally.

3. It usually has some form of self-similarity, perhaps approximate or statistical.
4. Its fractal dimension (Hausdorff dimension) is usually higher than its Euclidean dimension.
5. In most cases of interest, a fractal is defined in a very simple way, perhaps, recursively.

Most fractal compression algorithms require the segmentation of the image into blocks as stated in section 1. The following section concentrates on this issue.

III. Types Of Segmentations

In this section, we survey the most commonly used segmentation algorithms used in fractal compression. These algorithms serve as a stepping stone in our proposed algorithm.

A. Block based FIC

In fractal compression methods that employ block segmentation (BS), any image of dimensions (N x M) is decomposed into partitions called blocks. These blocks constitute the block domain pool D which contains the block domain elements D_i of dimensions (a x b). It is represented as: D_i; where $0 < i \leq N_1$; and N_1 in this case is the number of the domain blocks and is given by:

$$N_1=(NxM)/(axb) \quad (1)$$

To compress an image, it is decomposed again into block ranges R_j with size (a' x b') which is less than the size of D_i. Each block range R_j has its own transform τ_j, which maps the block range to its corresponding block domain. This transform is a characteristic feature of the block range. This transform has two parts as follows [1]:

$$\tau_j=T_j+S_j \quad (2)$$

Where T_j, is the gray scale transform, and S_j is the geometrical transform. The set containing all transforms of all block ranges R_j's of the whole image is called $\boldsymbol{\tau}$. It can be expressed as follows [1]:

$$\boldsymbol{\tau} = \sum \tau_j \quad (3)$$

where $0< j\leq$ (N/a' x M/b') and refers to the number of the block ranges in usage.

In the FIC scheme, a search is performed for each block range R_j, in the domain pool D to find the best match with the appropriate transform τ_j.

B. Region segmentation

The difference between this method and the previous one lies in the method of segmentation. Instead of BS segmentation, the image is segmented into regions of arbitrary shapes [1,2]. Each region is decomposed alone to construct the domain pool as in the BS method.

This method may be useful for object compression. The object to be compressed has a non uniform shape. On the other hand, the BS method is appropriate for uniform shape objects [1, 5].

IV. The Proposed Algorithm

This section gives the proposed simplified fractal compression algorithm. This algorithm is based on the simple segmentation of the image into block regions without a need for geometrical transformations applied to these regions when the search for best match is carried out. The simple proposed algorithm is a simplification of the strategy implemented in most fractal algorithms that avoid the so many geometrical transformations in the search process. The steps of the proposed algorithm are summarized as follows;

1- For an image of size N x M, calculate (N/a), and (M/b) where a and b are the dimensions of the block domain. If N/a and/or M/b are non integers, perform a zero padding process to obtain integer ratios.

2- Decompose this image into block domains D (i , j) with size (a x b), and a=b=8 or 16 as shown in Fig.1, Where, i=1:N/a , and j=1:M/b.

3- Select the reference block from each region by a comparison process between each block and all other blocks in the region. The block which gives distance less than a certain threshold with all other blocks in that region is selected as the reference block for the region. This process is repeated for all regions, as shown in Fig. 2.

D(1,1)	D(1,2)	D(1,3)	D(1,4)	…
D(2,1)	D(2,3)	D(2,3)	D(2,4)	…
D(3,1)	D(3,2)	D(3,3)	D(3,4)	…
D(4,1)	D(4,2)	D(4,3)	D(4,4)	…
D(5,1)	D(5,2)	D(5,3)	D(5,4)	
D(6,1)	D(6,2)	D(6,3)	D(6,4)	
D(7,1)	D(7,2)	D(7,3)	D(7,4)	
. .	. .	. .	. .	… …

Fig. 1, Block segmentation and block reference searching.

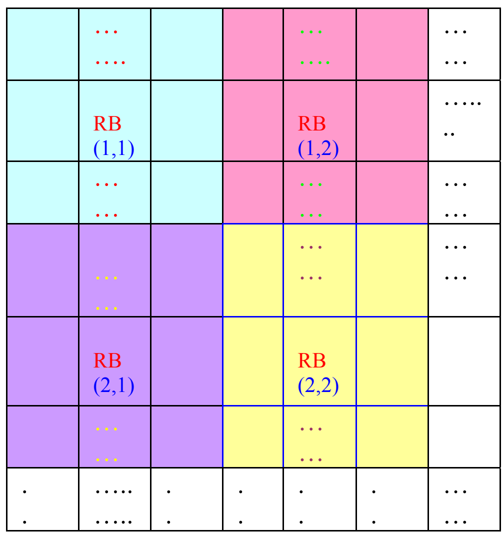


Fig. 2, Reference blocks in each region.

4- Formulate the domain pool from the reference blocks in each region as follows:

$$[D_1, D_2 \ldots D_{i1} \ldots D_{NRB}] = [RB(1,1), RB(1,2), \ldots RB(i,j), \ldots RB(N/a, M/b)] \quad (4)$$

Where NRB is the Number of Reference Blocks, and RB(i,j) is the reference block of region (i,j). The domain pool is used for storage or transmission.

5- Decompose each reference block domain into small blocks called reference ranges R_{ref} of size (a' x b'), similar to that of the range blocks. These blocks are used also for storage or transmission directly without processing.

6- Compute the number of pixels of the reference blocks (Ref);

$$Ref = NRB \times (a \times b) \quad (5)$$

7- Decompose the image to be compressed into block ranges R with size (a' x b') each.

8- For each block range R_i, a search process is conducted for its best match in its region first. If no match is achieved, a search in other regions is conducted.

9- Find the indices of the reference block achieving the best match with the range block. These indices are called (d_x , d_y).

10- If no match is achieved in all regions, the average of the range block is stored or transmitted as a representative of this block.

11- Compute the CR and the PSNR of the decompressed image.

PSNR = 10 log ($\sum^{N,M}$ (original image pixels – decoded image pixels)) / (number of the original image pixels) (6)

V. Experimental Results And Explanation

In this section, several experiments are conducted to test the efficiency of the simplified proposed fractal compression algorithm. In the first experiment, the Pout image of dimensions (291x240) is used as a prototype for compression. It is first zero padded to the size of (304x240). Then, the steps of the proposed algorithm are performed.

Figs. 3 and 4 give the results for the pout image compression experiment. The original image is shown in Fig. 3-a. Results in Fig. 3 have (16x16) block domains and (4x4) block ranges. The threshold (Thsh) is varied in parts (b) to (e) from 0.01 to 2. The experiments are repeated with (8x8) block domains and (4x4) block ranges and the results are given in Fig. 4.

A similar experiment is repeated for the second prototype Saturn image which has the dimensions of (328 x 438). This image is zero padded to (336x448). Then the steps of the proposed algorithm are performed. The results are shown in Figs. 5 and 6. The original image is shown in Fig. 5-a. The threshold ranges from 0.01 to 10.

It is obvious that the algorithm doesn't search for the affine transforms of each range block. It directly searches for similarity between the range block and the reference block in a reference domain based on the difference of distance between them. The resulting compression ratios vary between 3 and 10, where the PSNR ranges from 12 to 35dB for the Saturn image. For the Pout image the CR changes from 3.5 to 7 and the PSNR from 13 to 31dB. The decoded image quality is acceptable especially at higher PSNR (PSNR > 20).

Figs. 7 and 8 show the relation between the PSNR and the CR for the two images.

As expected, the PSNR decreases by increasing the threshold of matching which means increasing the distortion in the decoded image and this is a logical relation.

The curves which show the relation between the PSNR and the threshold are shown in Figs. 9 and 10. The resulting decoded image quality is acceptable with a CR =10 and 7 in the two cases.

This algorithm mainly gives a simplification in the encoding process. The time consumed to encode and decode the pout image by the proposed algorithm is 7.8 seconds for 16 x 16 blocks and 22.3 seconds for 8 x 8 blocks. For the Saturn image, it takes 16 seconds in the case of 16x16 blocks and 46 seconds in the case of 8x8 blocks. The experiments are carried out on a 1.78 Hz processor. The proposed algorithm is very fast if we compare this time with times consumed by many other fractal methods which may be thousands of seconds [2].

Table (1) is used to compare between the proposed algorithm and the square fractal image compression (SFIC) algorithm [3]. From this table, it is clear that the proposed algorithm can achieve a higher compression ratio in approximately half the time required for the SFIC algorithm if some losses in the compressed images are accepted.

VI. Conclusion

This paper has investigated fractal image compression from a different view point. Concentration in the paper is on how to implement fractal compression in as a simple manner as possible. So, geometrical transformations implemented in traditional fractal compression are eliminated. The obtained image quality using this simplified method is acceptable but with much more simplification.

TABLE 1
COMPARISON BETWEEN THE PROPOSED ALGORITHM AND SQUARE FRACTAL IMAGE COMPRESSION (SFIC) ALGORITHM.

Image	CR Proposed algorithm	CR SFIC	PSNR Proposed algorithm	PSNR SFIC	Time Proposed algorithm	Time SFIC
Saturn	10	9.8	25	31	15sec	30sec
Pout	7.2	5.2	22.3	32	8sec	15sec

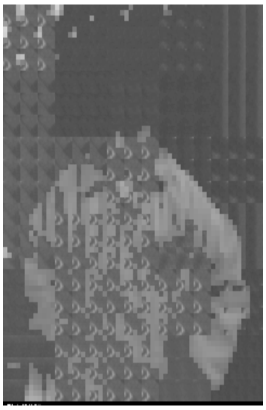
(a) The original image

(b)Thsh =0.01 CR=7
PSNR=31 a=b=16,
a'=b'=4

(c)Thsh=0.1,CR=6.1,
PSNR=30.95, a=b=16,
a'=b'=4.

(d)Thsh=0.5,CR=4.22,P
SNR=29.6, a=b=16,
a'=b'=4.

(e) Thsh=1,CR=4
PSNR=26.6,a=b=16,
a'=b'=4.



(f)Thsh=2,CR=3.8
PSNR=21.5,a=b=16,
a'=b'=4.

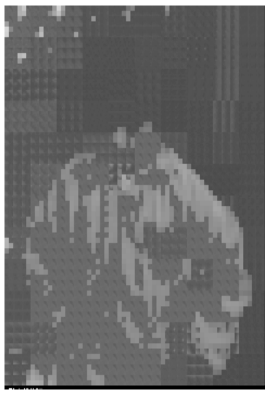
Fig. 3, Results of Pout image compression for 16x16 domain blocks.

(a)Thsh=.01,CR=2.6
PSNR=26.7 a=b=8
a'=b'=4.

(b)Thsh=0.1,CR=3.2989
PSNR=20.6,a=b=8,
a'=b'=4.

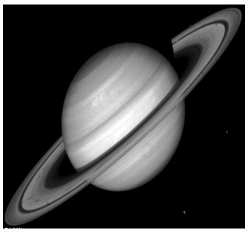
(c)Thsh=0.5,CR=2.1
PSNR=29.2,a=b=8,
a'=b'=4.

(d)Thsh=1,CR =5.2
PSNR=30.5,a=b=8,
a'=b'=4.

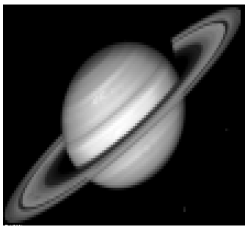


(e) Thsh=2, CR=7.2
PSNR=34.6,a=b=8,
a'=b'=4.

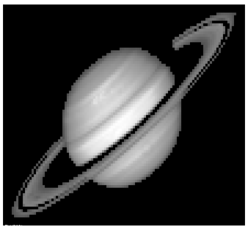
Fig. 4, Results of Pout image compression for 8 x 8 domain blocks.



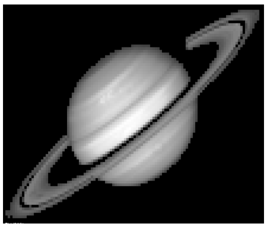
(a) The original image.



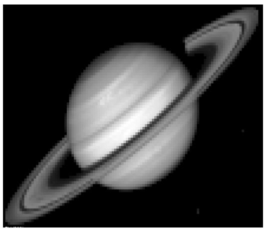
(b)Thsh=0.01,CR=9.86, PSNR=35.7,a=b=16,a'=b'=4.



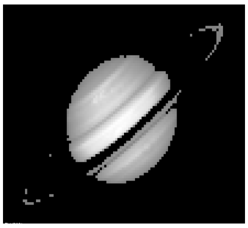
(c) Thsh =5, CR=9.2, PSNR=27.5,a=b=16, a'=b'=4.



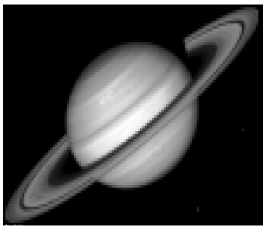
(d) Thsh=4, CR=9.34, PSNR=30.9, a=b=16, a'=b'=4.



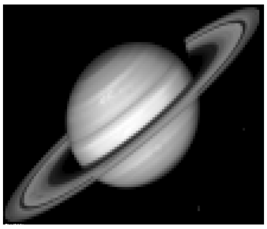
(e) Thsh=1, CR=9.71, PSNR=35.7, a=b=16, a'=b'=4.



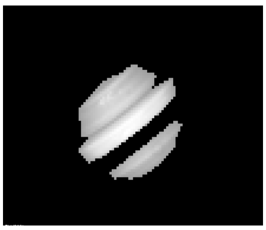
(f) Thsh=8, CR=8.7, PSNR=17.7, a=b=16, a'=b'=4.



(g) Thsh=0.1, CR=9.8, PSNR=35.7, a=b=16, a'=b'=4.

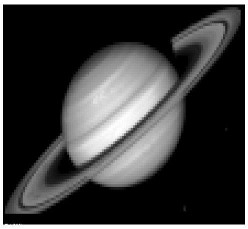


(h)Thsh=0.5,CR=9.75, PSNR=35.8, a=b=16, a'=b'=4.

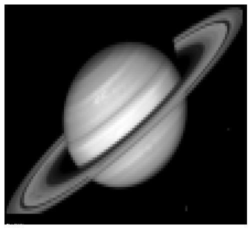


(k) Thsh=10, CR=8.4, PSNR=13.1, a=b=16, a'=b'=4.

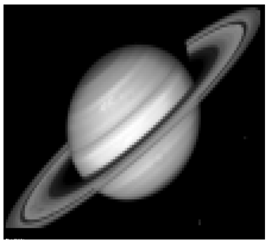
Fig. 5, Results of Saturn image compression for 16 x 16 blocks.



(a) Thsh=0.5, CR=9.99, PSNR=34.6, a=b=8, a'=b'=4.



(b) Thsh=0.1, CR=10, PSNR=34.6, a=b=8, a'=b'=4.



(c) Thsh=0.01, CR=10.1, PSNR=34.6, a=b=8 a'=b'=4.

Fig. 6, Results of Saturn image compression for 8 x 8 blocks.

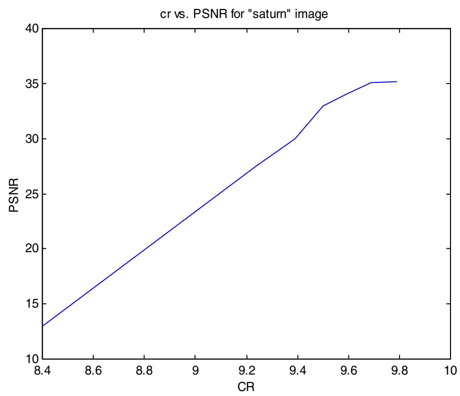


Fig. 7, CR versus PSNR for Saturn image.

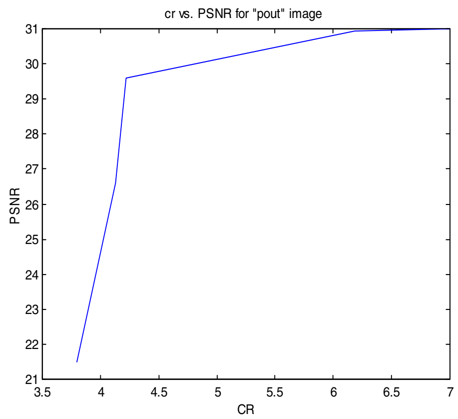


Fig. 8, CR versus PSNR for Pout image.

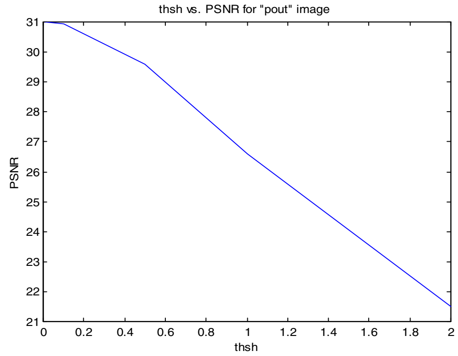


Fig. 9, PSNR versus threshold for Pout image.

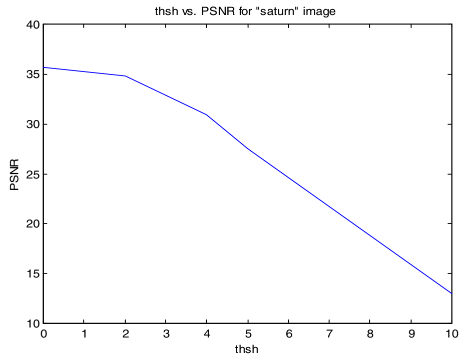


Fig. 10, PSNR versus threshold for Saturn image**.**

References

[1] Z. Erjun and L. Dan, "Fractal Image Compression Methods: A Review" Proceedings of the Third International Conference on Information, Technology and Applications, ICITA2005, pp.756-759, 2005.

[2] K. Eugene and G. Ong: "A Two-Pass Improved Encoding Scheme For Fractal Image Compression" Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV'06).

[3] X. He, H. Wang, Q. Wu, T. Hintz and N. Hur, "Fractal Image Compression On Spiral Architecture" Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV'06).

[4] M. Barnsley and A. D. Sloan, "A Better Way To Compress Images" BYTE, vol. 13, pp. 215-224, January 1988.

[5] D. Saupe and R. Hamzaoui, "Complexity Reduction Methods For Fractal Image Compression" Proceedings of the IMA Conference on Image Processing; Mathematical Methods and Applications, 1994.

[6] A. Jacquin, "Image Coding Based On A Fractal Theory of Iterated Contractive Image Transformations" IEEE Trans. on Image Processing, vol. 1, pp. 18-30, 1991.

[7] Y. Fisher, "Fractal Image Compression: Theory and Applications," Springer-Verlag, 1995.

[8] N. lu, "Fractal Imaging", Academic Press, New york, 1997.

[9] B. Wohlberg and G. de Jager, " A Review of The Fractal Image Coding Literature", IEEE Trans. on image processing, vol.8, No.12, December 1999.

[10] M. Barnsely "Fractals Everywhere" Academic Press, San Deigo, 1989.

[11] D. Saupe "A New View of Fractal Image Compression as Convolution Transform Coding" IEEE Signal Processing Letters, vol. 3, No. 7, July 1996.

[12] B. Wohlberg and G. de Jager, "A Review of the Fractal Image Coding Literature" IEEE Trans. on Image Processing vol. 8, No. 12, Dec. 1999.

[13] C. Dong, and M. Hirobe "Optimal Fractal Image Coding" Proceedings of IEEE Tencon, 1999.

[14] D. Jackson and S. Hannah, "A Comparative Analysis of Image Compression Techniques," Proceedings of The Twenty-Fifth Southeastern Symposium on System Theory, 513-517, March 7-9, 1993.

[15] D. Monro, S. Woolley, "Fractal image compression without searching" Proceedings of ICASSP, 1994.

[16] H. Chu and C. Chen "AN Efficient Decoding Scheme For Fractal Image Compression" Proceedings of the ICIP, 2000.

[17] W. C. Huang and L. W. Chang, "Predictive Subband Image Coding With Wavelet Transform," Signal Processing: Image Communication., vol. 13, pp. 171–181, 1998.

[18] H. Wang, M. Wang, T. Hintz, Q. Wu and X. He, "VSA Based Fractal Image Compression" Journal of WSCG, vol.13, No.1-3, pp.89-96, 2005.