Designing and Implementation of Efficient DCT Domain Based Fractal Image Compression Technique Using Quadtree Algorithm

Preeti Banerjee
Student :-Mtech Computer science
Mats University
Raipur, India

Deepak Kumar Xaxa
Dept. of computer science
Mats University
Raipur, India

Abstract - **Fractal image compression (FIC) is an image coding Technology based on the local similarity of image structure. It is broadly used in many fields such as image retrieval, image denoising, image authentication, and encryption. FIC, still, suffers from the high computational complexity in encoding. Although lots of schemes are purposed to speed up encoding time, they do not easily satisfy the encoding time or the reconstructed image quality needs.**

In this paper a new novel approach for fractal image compression based on DCT domain with Quadtree algorithm is proposed. In the proposed technique one domain block is considered for each range block and searched only for matched contrast scaling. Hence the outcomes fractal code does not contain coordinates of the matched domain block. This leads the advantage that the Quadtree algorithm can be here applied and the size of range block can be reduce as small as 2x2 pixels. Hence the quality of decoded image can be improved while the compression ratio can be maintained. The results obtained after implementation of the proposed fractal image compression technique shows that the encoding time required is very small as compare to conventional fractal image compression (CFIC) techniques. Moreover the PSNR and compression values obtained from proposed work are higher than CFIC.

Key Words:- Fractical image compression, Discrete cosine transform (DCT), quadtree encoding and decoding, PSNR, compression ratio.

1. INTRODUCTION

Fractal image compression (FIC) is an image coding technology based on the local similarity of image structure. It was proposed by Michael F. Barnsley in 1988 [1], and improved by Arnaud E. Jacquin in 1992 [2]. Jacquin's FIC scheme is called the baseline fractal image compression (BFIC), which uses the partitioned iterated function system (PIFS) to search the matching block pairs without human-computer interactions. Since then, FIC has a quick development and numerous FIC schemes have been published. Traditional image coding technologies encode an image by pixel-based and statistical methods, and FIC is an interesting attempt at the structure-based image coding. FIC has been used not only in image coding, but also in some interesting image problems [3]–[9] and pattern recognition problems such as facial recognition [10]. However, compared with traditional image coding technologies, fractal compression suffers from the high computational complexity in encoding.

Generally, one image is firstly partitioned into some square blocks in FIC, and then these square blocks compose a set called the pool. According to two kinds of size, an image is partitioned into two different pools. The pool composed by the blocks with larger size is called the domain pool, and the other pool is called the range pool. The cells in the range pool are the blocks to be encoded. The blocks in the domain pool are contracted to the same size as the range blocks, and then FIC takes the domain pool as a virtual codebook. The matching domain block for each range block needs to be searched. Exhaustive search of the matching block pairs costs too much time, which is one of the major difficulties in BFIC.

Although numerous schemes were proposed to speed up encoding in FIC, the encoding time is still too long so far. For example, the encoding time for a 512×512 image with 4×4 range blocks is more than 20 seconds in the DRDC scheme proposed by Riccardo Distasi et al in 2006 [18], and the encoding time for a 256×256 image is more than 2.8 seconds in the DUFC scheme proposed by Yi-Ming Zhou et al in 2009 [19]. Of course, it should be considered that the number of pair wise block-comparisons for a 512×512 image is 16 times more than the number of pair wise comparisons for a 256×256 image in BFIC with the same partition. Although these schemes should not be compared without the same computer environment, they still show that there is a lot of work to do for speeding up encoding in FIC.

In this paper a new novel approach for fractal image compression based on DCT domain with Quadtree algorithm is proposed. In the proposed technique one domain block is considered for each range block and searched only for matched contrast scaling. Hence the outcomes fractal code does not contain coordinates of the matched domain block. This leads the advantage that the Quadtree algorithm can be here applied and the size of range block can be reduce as small as 2x2 pixels. Hence the quality of decoded image can be improved while the compression ratio can be maintained.

2. FRACTAL IMAGE CODING

Conventional fractal coding starts from the portioning the input image into non overlapping range blocks of size, where, X is a predefined parameter. Then a set of domain block is created from original image, taking all square blocks of size with integer step L, in horizontal and vertical directions. Related to each no in domain pool, where new domain blocks are created with clock wise rotating it,

180 , and 270 , also these three and the original domain block all are mirrored. Here, in addition to the original domain block, there are seven new domain blocks, these new seven domain blocks are added to the domain pool, related to each range block the best domain block must be selected from domain pool and then the affine transform is calculated which maps the selected domain block to it with minimum distance. The mentioned distance amid a range block R, and a decimated domain block D together with n pixels is defined as follows:

$$E(R, D) = \sum_{i=1}^{n} (sd_i + o - r_i)^2 \qquad \dots(1)$$

That the best coefficient of S and O are [1]:

$$s = \frac{< R - \overline{R}.1, D - \overline{D}.1 >}{\| D - \overline{D}.1 \|^2} \qquad \dots(2)$$

$$o = \overline{R} - s\overline{D} \qquad \dots(3)$$

are inner product, range block, domain block, mean of R and mean of D respectively. Because of high computational cost of equation 2, it is convenient to search S across a pre sampled set of [0, 1] instead of calculating equation 2.

Along the matching process, the best found transformation only saved for range blocks which, have been mapped with a suitable error. The remaining range blocks are divide into four new smaller range blocks, and the matching process is restarted for them as a new step. For example if ranges initially have size of 16 x 16 pixels, the range blocks of the succeeding steps will have size of 8 x 8, 4 x 4 and 2 x 2 respectively, that leaves a four step algorithm. This is called Quadtree algorithm. Two approaches were used to reduce the encoding time in fractal coding algorithms. In a research strategy, finding out the domain pool is not necessary include all the possible domain blocks, only the high variance blocks are sufficient [5]. In other work like mentioned above the entropy measure was used instead of variance [6]. These methods are fast enough but the no search method is the best choice for real time fractal encoding. In the next section we will describe the proposed technique for fractal image compression.
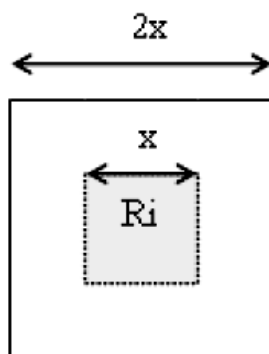

Figure (2.1) Position of range block and corresponding domain block.

3. Fractal image coding in DCT domain

Discrete cosine transform (DCT) has shown its capability of information compacting. Therefore it is widely used in image compression applications as compare to other transforms. The 2DCT transform of a L x L image $f(i, j), where\ i, j = 0, 1 \cdots L - 1$ is defined as follows

$$F(m,n) = \frac{2}{L} C_m C_n \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} f(i, j) \cos\left(\frac{(2i+1)m\pi}{2L}\right) . \cos\left(\frac{(2i+1)n\pi}{2L}\right) \qquad \dots(4)$$

Where   and

$$C_k = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

Therefore application of 2DCT transform on L x L image block generates L x L coefficients. If the size of image block is very large this process is very time consuming. Therefore it is often used to transform image blocks not on whole image. There are two strategies for fractal image coding process in DCT domain. The first is to apply DCT on the entire image and then perform portioning it into range blocks. This scheme is not very much useful, because it is difficult to find out a suitable quantization table related to human vision system. For example a smaller error on one coefficient adds an artificial sine wave to the whole image that can be seen easily in the smooth regions. Another problem is that the coefficient with small magnitude corresponds to edges of picture may be rounded to zero. Also as mentioned above applying DCT transform to whole image is very time consuming. The conservative way to solve this problem is to partition the image into blocks and then apply DCT to these blocks. These blocks become range blocks for fractal image coding.

4. PROPOSED METHODOLOGY

The proposed work mainly concentrates on the efficient fractal image encoding with good compression ratio and PSNR. The main steps of the proposed work of this paper represented in figure (4.1) with the help of the flow chart representation.
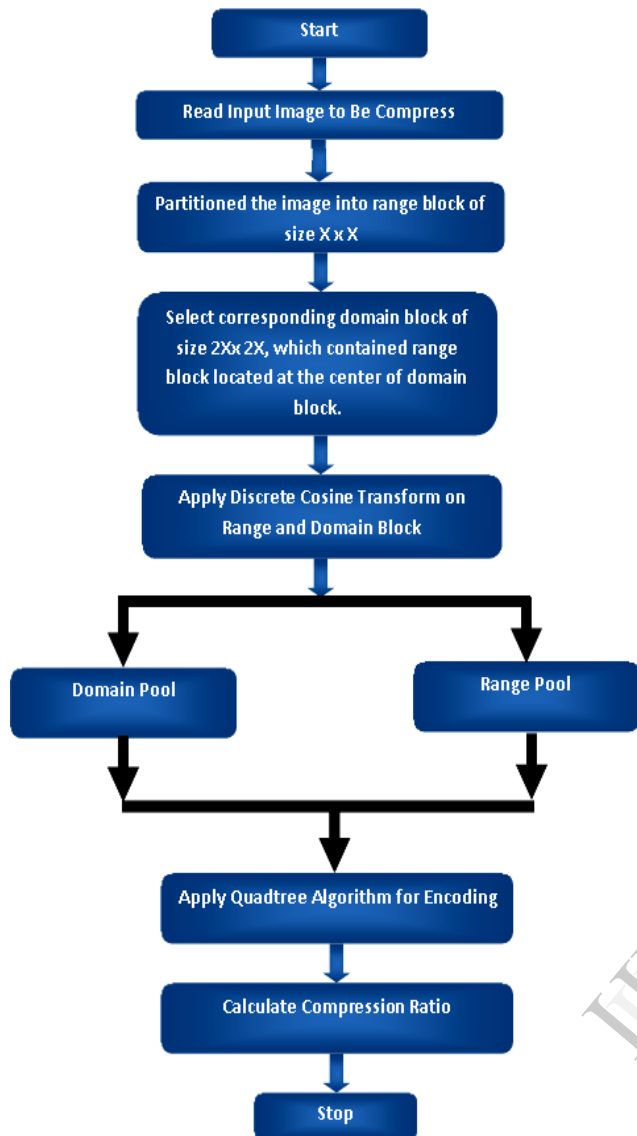
Figure (4.1) Proposed Fractal Image Compressor.

Figure (5.1)   Figure (5.1a)

Figure (5.2)   Figure (5.2a)

Figure (5.3)   Figure (5.3a)

5. Results and Discussions

The proposed technique has been successfully implemented in MATLAB 2012b. For the testing purpose three images have been used, they are shown from figure (5.1) to figure (5.3). After the compression and decompression process using proposed technique the resultant images are also shown in this section. For example Figure (5.1a) shows the decompressed image after the decompression of first image. For the proper result analysis consider fractal image compression parameter quality constant to 5.
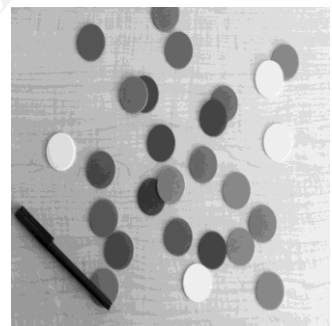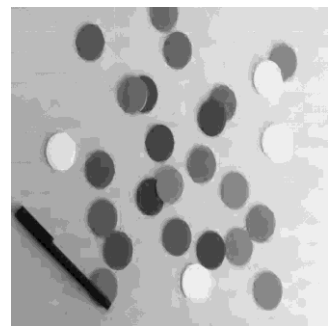
Now table 1, 2 and 3 shows the parameter obtained after compression and decompression of first, second and third input images respectively using proposed fractal image compression technique.

Table - 1

Image	Image Size	Image Size before compression in Bytes	Image Size After compression in Bytes	Image Size After decompression in Bytes	Encoding Time for compression	Compression Ratio	PSNR
First Input Image	512 x 512	2097152	161152	2097152	0.0771	13.0135	36.7927

Table – 2

Image	Image Size	Image Size before compression in Bytes	Image Size After compression in Bytes	Image Size After decompression in Bytes	Encoding Time for compression	Compression Ratio	PSNR
Second Input Image	512 x 512	2097152	161152	2097152	0.1021	10.8056	37.1629

Table - 3

Image	Image Size	Image Size before compression in Bytes	Image Size After compression in Bytes	Image Size After decompression in Bytes	Encoding Time for compression	Compression Ratio	PSNR
Third Input Image	512 x 512	2097152	161152	2097152	0.026	26.5543	38.1455

From the table 1, 2 and 3, it is clear that the proposed fractal image compression technique is not only able to provide higher PSNR and compression ratio, and simultaneously able to encode by taking very less time.

6. CONCLUSIONS

The over-long encoding time in fractal image compression is one of the major difficulties for its application. Although many researches were published to speed up encoding of FIC, the encoding time is still long, and in some schemes the reconstructed image quality becomes unacceptable. In this paper a new novel approach for fractal image compression based on DCT domain with Quadtree algorithm was proposed. In the proposed technique one domain block is considered for each range block and searched only for matched contrast scaling. Hence the outcomes fractal code does not contain coordinates of the matched domain block. This leads the advantage that the Quadtree algorithm can be here applied and the size of range block can be reduce as small as 2x2 pixels. Hence the quality of decoded image can be improved while the compression ratio can be maintained. From the results obtained shown in the table 1, 2 and 3, it is clear that the proposed fractal image compression technique is not only able to provide higher PSNR and compression ratio, and simultaneously able to encode by taking very less time.

REFERENCES

[1] M. F. Barnsley and A. E. Jacquin, "Application of recurrent iterated function systems to images," Proc. SPIE, vol. 1001, pp. 122–131, Nov. 1988.

[2] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," IEEE Trans. Image Process., vol. 1, no. 1, pp. 18–30, Jan. 1992.

[3] M. Pi, M. K. Mandal, and A. Basu, "Image retrieval based on histogram of fractal parameters," IEEE Trans. Multimedia, vol. 7, no. 4, pp. 597–605, Aug. 2005.

[4] J. H. Jeng, C. C. Tseng, and J. G. Hsieh, "Study on Huber fractal image compression," IEEE Trans. Image Process., vol. 18, no. 5, pp. 995–1003, May 2009.

[5] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," IEEE Trans. Image Process., vol. 12, no. 12, pp. 1560–1578, Dec. 2003.

[6] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," IEEE Trans. Image Process., vol. 15, no. 9, pp. 2669–2675, Sep. 2006.

[7] S. S.Wang and S. L. Tsai, "Automatic image authentication and recovery using fractal code embedding and image inpainting," Pattern Recognit., vol. 41, no. 2, pp. 701–712, 2008.

[8] S. G. Lian, "Image authentication based on fractal features," Fractals, vol. 16, no. 4, pp. 287–297, 2008.

[9] K. T. Lin and S. L. Yeh, "Encrypting image by assembling the fractalimage addition method and the binary encoding method," Opt. Commun.,vol. 285, no. 9, pp. 2335–2342, 2012.

[10] X. Tang and C. Qu, "Facial image recognition based on fractal image encoding," Bell Labs Tech. J., vol. 15, no. 1, pp. 209–214, 2010.

[11] B. Wohlberg and G. de Jager, "A review of the fractal image coding literature," IEEE Trans. Image Process., vol. 8, no. 12, pp. 1716–1729, Dec. 1999.

[12] H. Hartenstein, M. Ruhl, and D. Saupe, "Region-based fractal image compression," IEEE Trans. Image Process., vol. 9, no. 7, pp. 1171–1184, Jul. 2000.

[13] T. K. Truong, C. M. Kung, J. H. Jeng, and M. L. Hsieh, "Fast fractal image compression using spatial correlation," Chaos Solitons Fractals, vol. 22, no. 5, pp. 1071–1076, 2004.

[14] X. Wang, Y. Wang, and J. Yun, "An improved fast fractal image compression using spatial texture correlation," Chin. Phys. B, vol. 20, no. 10, pp. 104202-1–104202-11, 2011.

[15] Y. Fisher, E. W. Jacobs, and R. D. Boss, "Fractal image compression using iterated transforms," Image Text Compress., vol. 176, pp. 35–61, May 1992.

[16] W. R. Schwartz and H. Pedrini, "Improved fractal image compression based on robust feature descriptors," Int. J. Image Graph., vol. 11, no. 4, pp. 571-587, 2011.

[17] T. Kovacs, "A fast classification based method for fractal image encoding," Image and Vision Computing, vol. 26, no. 8, pp. 1129–1136, 2008.

[18] R. Distasi, M. Nappi, and D. Riccio, "A range/domain approximation error-based approach for fractal image compression," IEEE Trans. Image Process., vol. 15, no. 1, pp. 89–97, Jan. 2006.

[19] Y. Zhou, C. Zhang, and Z. Zhang, "An efficient fractal image coding algorithm using unified feature and DCT," Chaos Solutions Fractals, vol. 39, no. 4, pp. 1823–1830, 2009.

[20] C. He, S. Yang, and X. Huang, "Variance-based accelerating scheme for fractal image encoding," Electron. Lett., vol. 40, no. 2, pp. 1052–1053, 2004.

[21] H. N. Chen, K. L. Chung, and J. E. Hung, "Novel fractal image encoding algorithm using normalized one-norm and kick-out condition," Image Vis. Comput., vol. 28, no. 3, pp. 518–525, 2010.

[22] C. He, X. Xu, and G. Li, "Improvement of fast algorithm based on correlation coefficients for fractal image encoding," Comput. Simul., vol. 12, no. 4, pp. 60-63, 2005.

[23] J. Wang, Y. Liu, P. Wei, Z. Tian, Y. Li, and N. Zheng, "Fractal image coding using SSIM," in Proc. 18th ICIP, Sep. 2011, pp. 245–248.

[24] A. E. Jacquin, "Fractal image coding: A review," Proc. IEEE, vol. 81, no. 10, pp. 1451–1465, Oct. 1993.

[25] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE Signal Process. Lett., vol. 9, no. 3, pp. 81–84, Mar. 2002.