IMPROVEMENT ON FRACTAL IMAGE CODING BASED ON THE DIFFERENCE IMAGE

Huaqing Wang, Xiangjian He, Qiang Wu and Tom Hintz

Department of Computer Systems
University of Technology, Sydney
PO Box 123, Broadway 2007
NSW, Australia
Email: {huwang, sean, wuq, hintz}@it.uts.edu.au

ABSTRACT

In this paper, we propose an enhanced Fractal Image Compression (FIC) approach. In this method, the original image is represented by a provisional image and a difference image, which records the difference between the original and provisional image. With the less number of domain blocks required to do the best match search, the encoding time can be reduced by about 30% with a tolerance loss in the image fidelity. Furthermore, the idea behind this method has great potential to be utilized to improve other enhanced FIC.

1. INTRODUCTION

Fractal image compression is a relatively recent image compression method which exploits similarities in different parts of the image. For example, with a picture of a fern (Figure 1) one can see easily where these similarities lie: each fern leaf resembles a smaller fern. This is known as the famous Barnsley fern [1]. During more than two decades of development, the IFS (Iterated Function System) based compression algorithm stands out as the most promising direction for further research and improvement [2]. The basic idea is to represent an image by fractals and each of which is the fixed point of an IFS. An IFS consists of a group of affine transformations [3]. Therefore, an arbitrary image can be represented by a series of IFS codes, which consist of only three values each – an index to the block of similar information, a scale and an offset. In this way, a huge compression ratio 10000:1 can be expected [4]. In summary, there are diverse advantages offered by fractal coding algorithm, such as high decompression speed, high bit rate and resolution independence.

However, apart from the patent issue, the greatest disadvantage of fractal coding is the high computational cost of the coding phase. It makes the fractal coding inferior to other popular techniques (JEPG and wavelets, etc.) for the time being. For example, in the very first applicable fractal encoder [5], the given image was first divided into range blocks of same size, typically 4×4 or 8×8. Then for each range block, another square block on the same image with the most similar information to this range block needed to be found. This similar block was called a domain block, which was set to be four times larger than the given range block. For such a search, a single universal domain pool for all range blocks was defined by including all available blocks of the predefined size in the given image space. Each domain block in the domain pool needed to be scaled down by averaging the intensities of four neighboring pixels to the same size as range blocks, called a codebook block. The best similarity was measured by the least error of the least square root algorithm applied on the given range block and a certain codebook block. To fully investigate the image content for a best match between range and codebook blocks, some simple image transformation methods, such as rotation (0°, 90°, 180° and 270°), flipping (vertical and horizontal) and mirroring (vertical and horizontal), are usually applied on the domain blocks to enlarge the domain pool size by eight times. Therefore, in order to encode a given image of size 256×256 with the range block size 4×4 and domain block size 8×8, there will be 4096 range blocks and 496,008 possible codebook blocks. Therefore, the fractal encoding phase will at least include more than two billions times of least square root algorithm and last for hours or even days, which is obviously not acceptable.

Figure 1. A Barnsley fern.

Certainly, lot of research has been done to improve the fractal encoding [6-9]. Generally, the attempts to speed up the fractal encoding consist of modifying the following aspects [7]: the composition of the domain pool, the type of search used in block matching, or the representation and quantization of the transform parameters. However, these methods either require preprocessing on the given image, which is also of great computational complexity or are too specified to be compatible with other enhanced fractal encoders.

The approach proposed in the paper is based on the confirmed fractal coding characteristic that the fractal coding performs better on the images with less information, that is to say, images with smaller entropy, such as a sky image [6]. In the proposed method, provided the given image *O* is a grey level image, it is first encoded 'roughly' by searching in the domain pool consisting of merely a small number of neighboring domain blocks. The result image is denoted as *T*. Then a difference image *D* is generated to record the difference between the image *O* and *T*. Although the image *T* was a roughly-coded sample of image *O*, it still contains a significant amount of the detailed information in the original image *O* and leaves much less in the difference image *D* (see Figure 2), which then will be encoded using a fractal coder. Because of the minimum information in the image *D*, one may easily notice that the image *D* is particularly favorable to the fractal coding algorithm according to the rule stated previously. So the encoding time and also the number of bits required to restore *D* can be decreased dramatically but still keeps the fidelity at a satisfactory level. As a result, the original image *O* is recorded by the fractal codes of two images: the images *T* and *D*. Due to the much smaller domain pool size, the experimental results have shown that compared with encoding the original image *O* directly, the calculation required to encode both image *T* and *E* decreases by about 30%. Furthermore, the fidelity of the decoded image by the proposed method is remains at a comparable level at the same compression ratio.

Figure 2. Images *O*, *T* and *D* (left to right)

The organization of this paper is as follows. The basic fractal encoder and decoder are introduced in Section 2. The proposed new approach is presented in Section 3. Section 4 demonstrates the experimental results. Conclusions and discussions are given in Section 5

2. BASIC CONCEPTS OF FRACTAL IMAGE COMPRESSION

In the following section, the basic concepts of fractal image compression on the traditional square structure would be introduced. Before delving into details, there are some highlights of fractal image compression.

- It is a promising technology, though still relatively immature.
- The fractals are represented by Iterated Function Systems (IFSs).
- It is a block-based lossy compression method.
- Compression has traditionally been slow but decompression is fast.

2.1 Theory and Math Background

The fundamental principle of fractal image compression consists of the representation of an image by an iterated function system (IFS) of which the fixed point is close to that image. This fixed point is so called '***fractal***' [6]. Each IFS is then coded as a contractive transformation with coefficients. Banach's fixed point theorem guarantees that, within a complete metric space, the fixed point of such a transformation may be recovered by iterated implementation thereof to an arbitrary initial element of that space [10]. Therefore, the encoding process is to find an IFS whose fixed point is similar to the given image. The usual approach is based on the ***collage theorem***, which provides a bound on the distance between the image to be encoded and the fixed point of an IFS (more details please refer to [6] chapter 2). A suitable transformation may therefore be constructed as a 'collage' from the image to itself with a sufficiently small 'collage error' (the distance between the collage and the image) guaranteeing that the fixed point of that transformation is close to the original image.

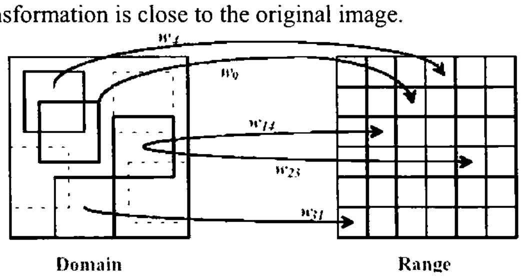


Figure 3. Each range block is constructed by a transformed domain block.

In the original approach devised by Barnsley, this transformation was composed of the union of a number of affine mappings on the entire image [3]. While a few impressive examples of image modelling were generated by this method (Barnsley's fern, for example), no automated encoding algorithm was found. Fractal image compression became a practical reality with the introduction by Jacquin of the partitioned IFS (PIFS) [5], which differs from an IFS in that each of the individual transformation operates on a subset of the image, rather than the entire image. Since the image support is tiled by 'range blocks', each of which is mapped from one of the 'domain blocks' as depicted in Fig. 3, the combined mappings constitute a transformation on the image as a whole. The transformation minimizing the collage error within this framework is constructed by individually minimizing the collage error for each range block, which requires locating the domain block which may be made closest to it under an admissible block mapping. For each range block, this transformation is then represented by the index to the matching domain block together with the block mapping parameters minimizing the collage error for that range block.

2.2 Basic Fractal Image Encoder

The encoder has to solve the following problem: for each range block R the best approximation

$$R \approx sD + oI \tag{2.1}$$

needs to be found, where D is a codebook block transformed from a domain block to the same size as R. The coefficients s and o are called ***scaling*** and ***offset***. We work out this problem with the Euclidean norm. That is, to minimize

$$E(D, R) = \min_{s,o} \|R - (sD + o\mathbf{1})\| \tag{2.2}$$

we can use the well known method of least squares to find the optimal coefficients directly as follows.

Given a pair of blocks R and D of n pixels with intensities $r_1, \ldots, r_n$ and $d_1, \ldots, d_n$ we have to minimized the quantity

$$\sum_{i=1}^{n} (s \cdot d_i + o - r_i)^2 \tag{2.3}$$

The best coefficients s and o are

$$s = \frac{n\left(\sum_{i=1}^{n} d_i r_i\right) - \left(\sum_{i=1}^{n} d_i\right)\left(\sum_{i=1}^{n} r_i\right)}{n \sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2} \tag{2.4}$$

and

$$o = \frac{1}{n}\left(\sum_{i=1}^{n} r_i - s \sum_{i=1}^{n} d_i\right) \tag{2.5}$$

With s and o given the square error is

$$E(D, R)^2 = \frac{1}{n}\left[\sum_{i=1}^{n} r_i^2 + s\left(s \sum_{i=1}^{n} d_i^2 - 2\sum_{i=1}^{n} d_i r_i + 2o \sum_{i=1}^{n} d_i\right) - o\left(on - 2\sum_{i=1}^{n} r_i\right)\right] \tag{2.6}$$

If the denominator in equation 2.4 is zero then

$$s = 0 \tag{2.7}$$

and

$$o = \sum_{i=1}^{n} r_i / n \tag{2.8}$$

In summary the baseline fractal encoder with fixed block size operates in the following steps.

1. **Image segmentation.** Segment the given image using a fixed block size, for instance, 4×4. The resulting blocks are called ranges R_i.
2. **Domain pool and codebook blocks definition.** By stepping through the image with a step size of 1 pixel(s) horizontally and vertically create a set of domain blocks, which are four times as the size of range blocks. By averaging the intensities of four neighboring pixels each domain blocks shrinks to match the size of the ranges. This produces the codebook blocks D_i.
3. **The search of best s and o.** For each range block R_i an optimal approximation $R_i \approx sD_i + oI$ in the following steps:

a) For each codebook block D_i compute an optimal approximation $R_i \approx sD_i + oI$ in three steps:

i. Perform the least squares optimization using formulas 2.4 and 2.5, yielding a real coefficient scalar s and an offset o.

ii. Quantize the coefficients using a uniform quantizer.

iii. Using the quantized coefficients s and o compute the error $E(R_i, D_i)$.

b) Among all codebook blocks D_i find the block D_k with minimal error

$$E(R_i, D_k) = \min_i E(R_i, D_i).$$

c) Output the code for the current range block consisting of indices for the quantized coefficient s and o and the index k identifying the optimal codebook block D_k.

2.3 Fractal Decoder

The fractal decoder is a very simple one. The reconstruction of the original image based on fractal codes consists of the implementation of the transformations described in the fractal code book iteratively to an arbitrary initial image. Every time the result image from the previous transformation is used as the starting point for the next transformation. Usually the decoded image can converge to a good approximation of the original image within less than seven iterations. The procedure is shown as an example in Figure 4. The number of iterations implemented is displayed at the top-left corner of each image.

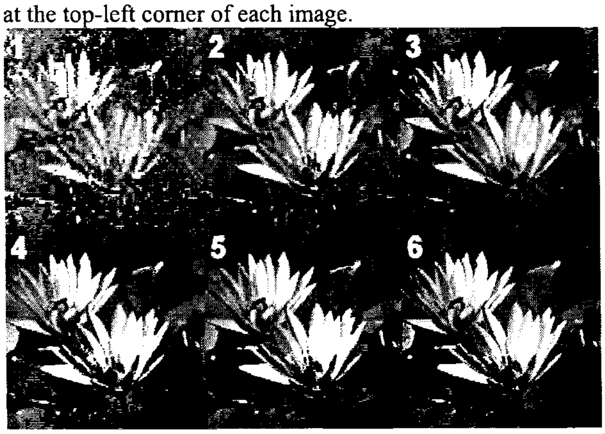


Figure 4. Decoded images after a certain number of iterations.

3. IMPROVEMENT ON FRACTAL IMAGE CODING BASED ON DIFFERENCE IMAGE

The fractal coding can achieve comparable or even superior compression result as JPEG and Wavelets, especially when the given image has a smaller entropy value. Take the image D in Figure 2 for example, the advantage can be easily demonstrated. Firstly, as the image itself has very less variety, most of the domain blocks would have similar information to each other. Therefore the domain pool size can be decreased dramatically by only including domain blocks of significant difference. This will then considerably reduce the coding time to find the best match for each range block. Meanwhile, as the pixel intensity values of the range block are already very similar to the ones in the domain blocks, the number of bits used to record the fractal codes are also can be reduced to some extent without introducing unacceptable distortions.

In application, provided the given image O is a grey level image, it is first encoded 'roughly' by searching in the domain pool consisting of merely a small number of neighboring domain blocks. The result image is denoted as T. Then a difference image D is generated to record the difference between the image O and T by the following formula:

$$D_{ij} = \frac{O_{ij} - T_{ij}}{2} + 128, \qquad (3.1)$$

in which, the D_{ij}, O_{ij} and T_{ij} are all the intensity values of the pixel on the position (i, j) of images D, O and T respectively. Afterwards, the image D is compressed by the fast fractal coding mentioned previously. Up to this point, the encoding phase has been completed: the original image O now is presented by two sets of codes rather than a single set when encoding it directly: one set to code the image T and the other for the difference image D.

The compressed file size with the two sets of codes is not necessarily larger than the single set when the given image is encoded using normal fractal coding algorithm. As introduced in the section 2, each range block is recorded by three coefficients: an index to the domain pool, a scale and an offset. For the proposed approach, the bits required for the index in both sets are much less than those required in the original methods. Additionally, the bits used for scales and offset can also be adjusted to help obtain variable compression ratios while retain the fidelity at a tolerable level.

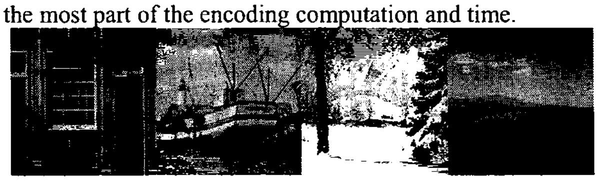
The decoder of the proposed approach has little modification. The two sets of codes are used to reconstruct the temporary image T and the difference image D following the same iterative procedure as the original decoding. Then the original image is approximated by the following formula:

$$O_{ij} = (D_{ij} - 128) \times 2 + T_{ij}, \qquad (3.2)$$

where each denotation is the same as in the equation 3.1.

4. EXPERIMENTAL RESULT

Four test images are all 256×256 one channel grey level images as shown in Figure 5. As for the experiments using the normal fractal coder, each 4×4 sized range block is encoded by storing the following information: the index to the best domain block (12 bits), the scale coefficient (7 bits) and offset coefficient (5 bits). As a result, each range block is coded by 12+7+5=24 bits. The domain block pool is designed to be the spatially closest 2^{12}=4096 domain blocks available to any given range block. As a result, during the normal fractal coding phase, the least root algorithm and the associated comparison to obtain the least error will be implemented by $4096 \times (256/4)^2 = 2^{24}$ times, which occupies the most part of the encoding computation and time.



(a) (b) (c) (d)

Figure 5. Test images. (a) Building, (b) Boat, (c) House, (d)Scene

On the other hand, in the proposed approach the test images are firstly encoded with range blocks of 4×4 size by the following information to obtain a temporary image: the index to the best domain block (5 bits), the scale coefficient (3 bits) and offset coefficient (4 bits). After the difference image is derived from the original image and the temporary image, it is also encoded by the encoder with the same three coefficients but with different bits assignment: the index 5 bits, the scale 3 bits and the offset 4 bits. Therefore, the compressed file size remains the same: 24bits/range block. However, the computational complexity decreases significantly. The numbers of least root algorithm implemented and its associated comparison to get the least error now are as following:

$$(2^5 \times 4096 + 2^5 \times 4096) = 2^{18},$$

which is merely the 1/64 of the original method. Accordingly, the encoding time can be reduced considerably. The following table summarizes the encoding time and PSNR value using both the original and proposed methods. It can be easily seen that compared with the original method, the encoding time is reduced by about 30% and the PSNR value still remains comparable.

Table 1. Experimental result summary

Images	Original FIC		Enhanced FIC	
	Time(sec.)	PSNR(dB)	Time(sec.)	PSNR(dB)
Building	49.3	23.37	34.3	24.21
Boat	51.1	26.43	35.6	25.93
House	50.2	22.89	34.3	22.77
Scene	51.2	28.74	36.4	26.98

5. CONCLUSIONS AND DISCUSSION

In this paper, we propose a novel approach to enhance the fractal coding by reducing the domain pool size with the assistance of a temporary image and a difference image generated during the encoding. The experimental results show that the proposed approach can reduce the encoding time by 30% with a tolerable loss in image fidelity. Meanwhile, the proposed method has great potential to be adopted onto other enhanced fractal coding methods to further improve the performance. The authors' direction of interests is currently in the integration of fractal image coding and a novel image structure – spiral architecture. Preliminary research on this topic can be found in [11-13]

6. REFERENCE

[1] Barnsley, M 1988, Fractal Everywhere, New York: Academic.

[2] Barnsley, M & Demko, S 1985, 'Iterated Function Systems and the Global Construction of Fractals', Royal Soc., vol. A399, London, p. 243~75.

[3] Barnsley, M & Hurd, LP 1993, Fractal Image Compression, AK Peters. Ltd.

[4] Barnsley, M & Sloan, AD 1988, 'A better way to compress images', BYTE, January, pp. 215-23.

[5] A. E. Jacquin, "Fractal image coding: a review," *Proceedings of the IEEE*, vol. 81, pp. 1451-1465, 1993.

[6] Y. Fisher, *Fractal Image Compression: Theory and Application*. New York: Springer-Verlag New York, Inc., 1995.

[7] B. Wohlberg and G. d. Jager, "A Review of the Fractal Image Coding Literature," *IEEE Transaction on Image Processing*, vol. Vol. 8, 1999.

[8] 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, 2006.

[9] Y. Iano, d. S. Silvestre, and A. L. M. Cruz, "A Fast and efficient Hybrid Fractal-Wavelet Image Coder," *IEEE Trans. Image Process.*, vol. 15, No.1, pp. 98-105, 2006.

[10] Kreyszlg, E 1978, Introductory Functional Analysis with Applications, New York: Wiley.

[11] H. Wang, X. He, Q. Wu and T. Hintz, "A new approach for fractal image compression a virtual hexagonal structure", International Conference on Pattern Recognition, Hong Kong, Aug. 2006

[12] 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

[13] H. Wang, Q. Wu, X. He, and T. Hintz, "Fractal Image Compression on Virtual Spiral Architecture," International Conference on Image Processing, Atlanta, USA, Oct., 2006

Networking

Wireless Systems