

A Fractal-based Hybrid Image Coding System

Zhou WANG and Yinglin YU
Institute of Radio and Automation
South China University of Technology
Guangzhou 510641, China

ABSTRACT

The interest in image compression by means of fractal techniques has been steadily growing in recent years since Jaquin proposed a fully automatic image compression algorithm called Fractal Block Coding (FBC). In this paper, we introduce a hybrid system where only edge region of the image is coded by fractal transform and the smooth region is modeled by other techniques. The non-fractal coded region is used to help the coding of fractal coded region, so that the fractal coded region can be represented by fewer bits. Experiments show that the overall coding efficiency is improved.

1. INTRODUCTION

The interest in image compression by means of fractal techniques has been steadily growing in recent years. A fully automatic image compression algorithm for real world gray level images which was referred to as Fractal Block Coding (FBC) [1-2] was proposed by Jacquin in 1989. Since then, almost all known research has been based upon Jacquin's method which has been analyzed, extended and refined e.g., [3-9]. The basic FBC scheme can be summarized as follows:

The original image is partitioned into non-overlapping square blocks called range blocks (R_i 's). For each R_i , the fractal encoder select a domain block D_i from a domain pool which approximate R_i through an affine mapping. Typically, R_i 's are of equal size ($B \times B$) and D_i 's are twice size of R_i 's ($2B \times 2B$). The domain pool can be obtained by sliding a window of size $2B \times 2B$ across the original image. The window is first located with its upper left corner at the upper left corner of the original image. It then moves from one position to the next by steps of typically $B/2$, B or $2B$ pixels horizontally to the right or vertically to the downwards. The mapping from D_i to R_i combines a geometrical transformation g_i and a luminance transformation l_i . g_i consists of a spatial contraction, a rotation and a position shift that maps D_i to the position of R_i . l_i is a linear mapping:

$$l_i(z) = s_i z + o_i \quad (1)$$

where z denotes pixel intensity, s_i and o_i are called scale factor and offset respectively [3] which are selected to minimize the Mean Square Error (MSE) between the transformed D_i and R_i . For the purpose of convergence, $s = \max(|s_i|) (i=1,2,\dots,N) < 1$ (In practice, s can be larger than 1 [3]). When decoding, the mappings, iteratively applied to

any initial image, generates a unique attractor (reconstructed image).

FBC is compatible with other image coding techniques. This means we can code only part of an image with fractal transform and use other image coding techniques to model the remaining part. If the fractal coded region or non-fractal coded region can make use of the information from the other side, then the coding efficiency for them may be improved. Fig. 1 shows a general diagram for this kind of hybrid image encoding-decoding system.

To develop such a hybrid system, we employ a modified FBC algorithm called Fractal Block Coding in Residue Domain (FBCRD) in this paper which is proposed by the authors in [5]. In FBCRD, neither range nor domain blocks are original blocks directly extracted from the original image but both are generated by subtracting their own block means from the original blocks. Since the block means for both domain and range blocks equal to 0, the luminance transformation (1) becomes $l_i(z) = s_i z$. Compared with basic FBC, FBCRD needs some more bits to code block means. However, the bits for o_i 's are saved. The overall coding efficiency of basic FBC and FBCRD is basically the same while FBCRD allows us to generate a reconstructed image in about 3 iterations, much less than 8-10 iterations of basic FBC.

We use FBCRD for two reasons. First, FBCRD holds their property of fast decoding. A fast decoding structure make it usable in real time system [6]. Second, block means instead of o_i 's are employed by FBCRD based method which can be predicted by the block's neighborhood region. Then the information of non-fractal coded region can be used in the prediction so as to help the coding of fractal coded region.

This paper is organized as follows: The hybrid system is introduced in Section 2. Section 3 further demonstrates the encoding and decoding procedures. Some concluding remarks are made in Section 4.

2. THE HYBRID SYSTEM

The hybrid image coding system combines several image coding and image processing techniques which can be summarized as follows:

1) The original image is partitioned into smooth region H and edge region G. A lossless data compression algorithm is used to code the classification result.

- 2) The smooth region H is modeled by a set of dc and linear approximations.
- 3) The block means of the edge blocks which compose the edge region G are predicted by its surrounding blocks and then be quantized and coded.
- 4) The edge blocks are coded using FBCRD method.

We use an example here to illustrate our algorithm clearly where the original image is "Lena" 512×512, 8bpp gray level image which is shown in Fig. 2.

2.1 Image Partition

We use *Sobel* operator to detect edges of the original image and achieve an edge image which is shown in Fig. 3. Then the edge image is partitioned into non-overlapping equal sized blocks (8×8, B=8). A block may hold many edge pixels or hold only a few or no edge pixel. A threshold of the number of edge pixels is used to determine whether the block is an edge block or a smooth block. All the edge blocks compose G part of the image and the remaining smooth blocks compose H part of the image.

The classification result which is shown in Fig. 4 should be coded using a lossless method. Here we choose an adaptive arithmetic coding method which is introduced in [10].

2.2 Approximation of smooth regions

To improve coding efficiency, we use a region growing algorithm to connect adjacent smooth blocks, so that the connected blocks can be modeled by one set of codes instead of one set for each block. To avoid the region growing algorithm to connect too many blocks which make the connected region difficult to be coded, we limit the connected blocks to be within a local region (e.g., a local 64×64 block).

For each connected region, we first simply try to approximate it by a uniformly gray region with gray level equal to its average gray level (dc component). If the resulting MSE is less than a threshold, the average value is quantized and stored. Otherwise, we use a plane $z=ax+by+c$ to approximate it.

2.3 Block Mean Prediction

As having been discussed in Section 1, the codes for H part may be used to improve coding efficiency of G part. Since block mean is employed by FBCRD based fractal transform, if we use the information of H part to predict block means of the blocks within G part, then the fractal transform for these blocks may be represented by fewer bits.

Each block has 8 direct surrounding blocks (see Fig. 5). We use the average of the block means of these 8 blocks to predict the block mean of the surrounded block. Note that a block in G part may not have all of its 8 direct surrounding blocks in H part. Actually, some may have only 1 or even 0 direct surrounding block in H part. Therefore, it is impossible to directly predict block means of these blocks. We use an iterative procedure to solve this problem.

In the first step, the smooth blocks are covered using the approximation method discussed in 2.2. In step 2, for each uncovered block, if all of its 8 direct surrounding blocks have been covered, compute the average of the block means of these 8 blocks as a prediction of the mean of the uncovered block. The difference between the real block mean and the predicted block mean is stored. Then the block is covered with its real block mean. Step 2 is repeated until no uncovered block has all of its 8 direct surrounding blocks covered. Step 3, 4, 5, ... are almost the same as step 2 except that the uncovered blocks are allowed to be covered if they have 7, 6, 5, ... instead of 8 direct surrounding blocks.

Normally, the difference between the real block mean and the predicted block mean is a value near zero, hence can be coded using fewer bits. A statistic of the difference is shown in Fig. 6.

An image with its H part covered by the method discussed in 2.2 and G part covered by block means is shown in Fig. 7.

2.4 Fractal approximation of edge blocks

For each block in G part, similar to that in 2.2, we firstly simply try to approximate it by a uniformly gray block with gray level equal to its block mean. If the resulting MSE is less than a threshold, we do not need more bits to code it because its block mean has been coded in 2.3. Otherwise, we use FBCRD method to model it.

Here we mainly describe the difference between the method of basic FBCRD based system and that used in our hybrid system. In our hybrid system, since an edge block and smooth block classification procedure has been applied and only edge blocks may be coded by fractal transform, it is reasonable to exclude smooth blocks from the domain pool. To obtain the domain pool, a 2B×2B sliding window is first located with its upper left corner at the upper left corner of the image. It then moves from one position to the next by steps of 2B pixels horizontally to the right or vertically to the downwards. If the 2B×2B block within the window contains edge blocks, it is included in domain pool. Otherwise, it is excluded from the domain pool.

The reduction of domain pool leads to 2 improvements. First, since most of the encoding time of fractal encoding is used in searching for suitable domain blocks in the domain pool, smaller domain pool means less encoding time. Second, smaller domain pool allows us to use fewer bits to code the position shift for the geometric transformation g_i 's, resulting in high coding efficiency.

3. ENCODING AND DECODING

The codes generated by the hybrid system consists of several parts.

i) Classification information.

The classification information for each block is represented by 2 bits according to Table 1.

Table 1. Classification codes for a block

classification codes	belongs to	modeled by
00	H part	dc component of connected region
01	H part	linear approximation of connected region
10	G part	its block mean
11	G part	fractal transform

A 512×512 image has 4096 8×8 blocks, so 1024 bytes are needed to code the classification information. This 1024 bytes can be further compressed to about 700-800 bytes by a lossless adaptive arithmetic coding algorithm [10].

ii) Codes for dc and linear approximation

For a dc approximated region, we use 8 bits to code the dc component.

To represent a connected region using linear approximation means to use a plane $z=ax+by+c$ to fill the region. The square error between the plane filled region and the original region is

$$E = \sum (ax + by + c - z)^2$$

Under the condition of minimal E , we can achieve a , b and c by solve the equations $\partial E/\partial a=0$, $\partial E/\partial b=0$ and $\partial E/\partial c=0$. by $\partial E/\partial c=0$, we get

$$c = m - a\bar{x} - b\bar{y}$$

where m is the mean value (dc component) of the region, \bar{x} and \bar{y} are the averages of x and y within the region respectively. Then the plane becomes

$$z = ax + by + c = a(x-\bar{x}) + b(y-\bar{y}) + m$$

a , b and m are coded using 8bits each in our algorithm.

iii) Codes for block means of blocks in G part

For each block in G part, the prediction algorithm discussed in 2.3 makes it possible to code its block mean in fewer bits. If the absolute value of the difference between the predicted block mean and the real block mean is less than 16, we use 5 bits to code it. Otherwise, we use 8 bits to store the real block mean. 1 more bit is needed to make a classification. If p represents the proportion of the 5 bits coded blocks in all blocks, then the average number of bits needed to code a block mean is $5p+8(1-p)+1=9-3p$. Experiments show that it is normally about 7, 1 less than that needed to directly code the block mean.

iv) Codes for fractal transforms

The codes for a block's fractal transform includes its block mean (precoded in iii), symmetry, scale factor and position shift. Each symmetry needs 3 bits, the same as that in basic FBC [2]. We limit the maximum absolute value of scale factor to 1.6 and code each scale factor using 6 bits. As having been discussed in 2.4, the domain pool in our algorithm is smaller than that in normal FBC. This make it possible to code the position shift for each block using fewer bits. For a 512×512 image with equal sized 8×8 range blocks, equal sized 16×16 domain blocks and domain

block selection step equal to 16, the domain pool holds 1024 domain blocks. This means 10 bits are needed for each position shift. Since the size of the domain pool is smaller in our algorithm, only 9 bits are used to code each position shift in our system.

At the decoder, the classification information is the first to be decoded. Then the codes for dc and linear approximation are applied to fill H part of the image. The block means of the blocks within H part is computed which are used to predict block means of the blocks within G part through an iterative procedure as discussed in 2.3. The codes of the difference between predicted block means and real block means are used in the iterative procedure. In the end, the fractal transforms of the fractal coded blocks are iteratively applied to generate the reconstructed image.

To reduce blockness, a smooth algorithm is post processed, so that the quality of the reconstructed image is improved. Fig. 8 shows the resulting decoded image.

4. CONCLUSIONS

This paper presents a hybrid system which combine fractal coding with other image coding techniques. Several image processing and image coding techniques are used in the system including edge detection, region growing, dc and linear approximations, block mean prediction, arithmetic coding and a modified FBC algorithm called FBCRD. For each fractal coded block, FBCRD replace offset of the luminance transformation of the fractal transforms by its block mean. This modification on the one hand accelerates decoding procedure, while on the other hand, makes it easier to use non-fractal coded region to help the encoding of fractal coded region. Experiments show that the overall coding efficiency is improved.

5. ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China, Defense Preresearch Foundation and Natural Science Foundation of Guangdong Province.

6. REFERENCES

- [1] A. E. Jacquin, "A fractal theory of iterated Markov operators with applications to digital image coding", PHD Thesis, Georgia Institute of Technology, Aug. 1989.
- [2] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformation", *IEEE Trans. Image Processing*, vol. 1, no. 1, 1992, pp. 18-30.
- [3] E. W. Jacobs, Y. Fisher, R. D. Boss, "Image compression : a study of the iterated transform method", *Signal Processing*, vol. 29, no. 3, Dec. 1992, pp. 251-263.
- [4] Y. Fisher, *Fractal Compression: Theory and Application to Digital Images*, Springer-Verlag, New York, 1994.

Trans. Image Processing, vol. 1, no. 1, 1992, pp. 18-30.

- [3] E. W. Jacobs, Y. Fisher, R. D. Boss, "Image compression : a study of the iterated transform method", *Signal Processing*, vol. 29, no. 3, Dec. 1992, pp. 251-263.
- [4] Y. Fisher, *Fractal Compression: Theory and Application to Digital Images*, Springer-Verlag, New York, 1994.
- [5] Zhou Wang, Huaqiu Deng, Yinglin Yu, "Fractal block coding in residue domain", submitted.
- [6] Zhou Wang, Yinglin Yu, "A fast decoding structure for fractal block coding in residue domain", submitted.

- [7] Yudong Fang, Yinglin Yu, "A quick fractal image compression coding method", *Acta Electronica Sinica*, vol. 24, no. 1, Jan. 1996, pp. 29-34.
- [8] Zhou Wang, Yinglin Yu, "A novel fractal image coding approach", *Journal of China Institute of Communications*, accepted.
- [9] M. S. Lazar, L. T. Bruton, "Fractal block coding of digital video", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 3, Jun. 1994, pp.297-308.
- [10] I. H. Witten, R. M. Neal, J. G. Cleary, "Arithmetic coding for data compression", *Communications of the ACM*, vol. 30, no. 6, Jun. 1987, pp. 520-540.

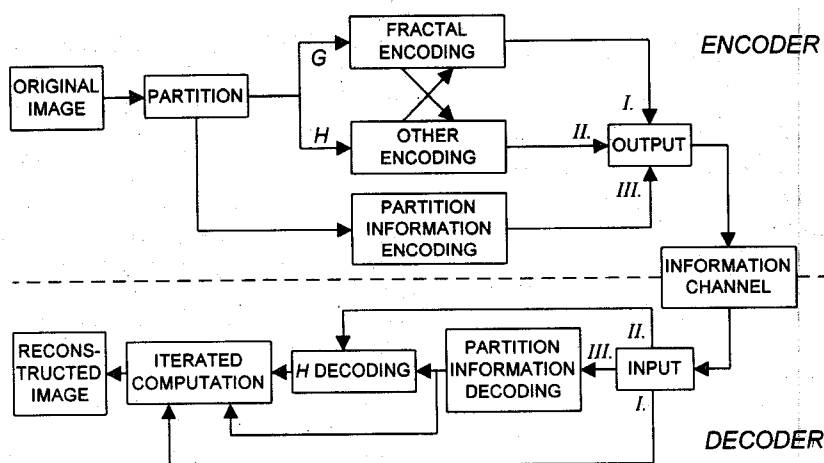


Fig. 1. General diagram of the hybrid system.



Fig. 2. Original image "Lena" 512x512, 8bpp.



Fig. 3. Edge image.



Fig. 4. Classification result. Black for smooth blocks and White for edge blocks.

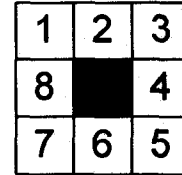


Fig. 5. A block and its 8 direct surrounding blocks.

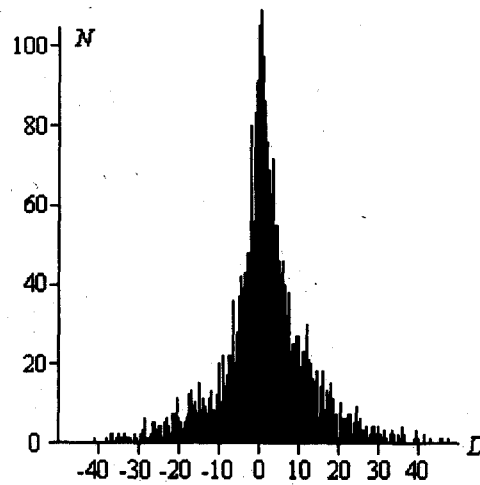


Fig. 6. Statistics of difference between real block mean and predicted block mean. D : difference; N : number



Fig. 7. The image after smooth region approximated by dc and linear approximation and edge blocks covered with their block mean



Fig. 8. The resulting decoded image. 30.2dB with compression ratio 37:1.