オーディオビジュアル複合情報処理　39－22
(2002. 12. 17)

拡張ブロックを用いた誤りに強いフラクタル画像符号化

世宮大輔　藤井俊彰　谷本正幸

名古屋大学工学研究科

〒464-8603 名古屋市千種区不老町
E-mail: semiya@tanimoto.nuee.nagoya-u.ac.jp

あらまし　フラクタル画像符号化手法において、その符号化データに誤りや欠落が生じると復号時の収束過程においてその誤りが広く分散、伝播してしまう。本稿ではレンジブロックを相補的な2つのグループに分割し、そのそれぞれを拡張してやることにより、復号側で2枚の復元画像を得られるようにし、両者を比較して誤り検出、修正を行う手法を提案する。本手法によって、データ誤り発生状況下において従来手法よりも2-3dB程度良好な復元画像が得られた。また、復号過程を工夫することにより、誤りの発生しない状況においても従来法とほぼ遜色ない復元画像を得ることができた。

キーワード　画像符号化、フラクタル、IFS、誤り耐性

Fractal Image Coding with Error Resilience by Using Extended Range Blocks

Daisuke Semiya, Toshiaki FUJII and Masayuki TANIMOTO

Department of Information Electronics,

Graduate School of Engineering, Nagoya University,

Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan

E-mail: semiya@tanimoto.nuee.nagoya-u.ac.jp

Abstract　Fractal image coding is weak in errors. When an error occurs in one block, the error propagates to the portion which refers to this block in the decoding process. In this paper, we propose a new coding method for error resilience by sending the image twice with the same amount of data as the single image. This is realized by using double-sized range blocks and a new criterion to determine the domain block. 2-3dB improvement of SNR is achieved in the reconstructed image when an error occurs, while keeping nearly the same SNR when an error doesn't occur.

Keyword　image compression, Fractal Image Coding, Iterated Function System, error tolerance

1. INTRODUCTION

As we know, highly compressed data is needed in the mobile communication, but they are strongly influenced by transmission errors. Therefore, high tolerance of errors is also needed under condition of mobile communication. Although image coding schemes using DCT like JPEG, MPEG are widely used, they don't have enough robustness. On the other hand, fractal image coding based on Iterated Function System (IFS) has been attracting much interest because of possibilities of drastic data compression. It achieves compression by using the self-similarity in an image. Using the self-similarity in the image, each block (range block) is mapped from double sized block (domain block), which is considered as the most suitable block to approximate the range block. The mapping process consists of dividing and scaling down a domain block, geometry transform, calculating a scaling parameter and a offset parameter, affine transform, and calculating RMSE (root mean square error) between a range block and a transformed block. Because of these features of fractal image coding, if a transmission error occurs in one block, it spreads out to other portions of image that refer to the block with the error. Therefore, we can say that fractal image coding doesn't have enough robustness. We found that the relation between the range block and domain block can also be applied well to the extended range blocks and domain blocks in most cases.

In this paper, we proposed a new scheme of fractal image coding with robustness according to these features. By our scheme, we can detect transmission errors of data compressed by fractal image coding scheme and correct the block of image automatically. We perform the image coding experiment based on the proposed scheme in order to verify the effectively of this scheme.

2. CONVENTIONAL SCHEME

2.1 Encoding Scheme

The basic idea of fractal image coding is described below. At the encoder, an image I is divided into non-overlapping square blocks (range blocks) r_i, $R \times R$ size of pixels. That is

$$\bigcup_{i=1}^{N_r} r_i = I, \quad r_i \cap r_j = \phi, \quad for \quad i \neq j \tag{1}$$

where N_r is the total number of range blocks.

For each range block r_i, we search for a larger block (domain block) D_k, of size of normally $2R \times 2R$ pixels in the image I, and a contractive mapping w_i, such that

$$r_i = w_i(D_{k_i}) \tag{2}$$

and the image I, which should be encoded, can be expressed as

$$I = \bigcup_{i=1}^{N_r} r_i = \bigcup_{i=1}^{N_r} w_i(D_{k_i}) = W(I) \tag{3}$$

i.e., I is the fixed point of W. A domain pool is a set of domain blocks defined as

$$D = \{D_j | j = 1, \ldots\ldots, N_D\} \tag{4}$$

where N_D is the total number of domain blocks. The mapping W usually has a fixed point I', which is only an approximation of I. For example,

$$\sum_{x,y=0}^{N-1} |I(x,y) - I'(x,y)| < e \tag{5}$$

for a real number e, where N is the size of the image I.

During the mapping process, the domain block D_k is low-pass filtered and subsampled from size $2R \times 2R$ down to size $R \times R$ (same as the size of r_i), and then processed by isometrical transform, contrast scaling and offset operations. Mathematically, the process can be written as

$$\hat{r}_i = d'_{k_i} = w_i(D_{k_i}) = \Psi_i(\Phi_i(D_{k_i} - \mu_{k_i})) = \alpha_i \Phi_i(D_{k_i} - \mu_{k_i}) + \beta_i \tag{6}$$

where Φ_i represents the spatial contraction and isometrical transform, Ψ_i represents contrast transform, and μ_k is the average of the domain block D_k. And α_i and β_i represent contrast scaling and luminance offset, such that

$$\alpha_i = \frac{R^2 \sum r_i d'_{k_i} - (\sum r_i)(\sum d'_{k_i})}{R^2 \sum d'^2_{k_i} - (\sum d'_{k_i})^2} = \frac{\sum r_i d'_{k_i}}{\sum d'^2_{k_i}}, \tag{7}$$

$$\beta_i = \mu_i \tag{8}$$

where μ_i is the average of the range block r_i. When a set of α_i and β_i is obtained, RMSE (root mean squared error) e_i is calculated by

$$e_i = \sqrt{\frac{1}{R^2} \sum \| r_i - d'_{k_i} \|^2} \tag{9}$$

The e_i is calculated for N_p of domain blocks around the range block, where N_p is the number of searching domain blocks. Then a code for the range block which achieves the minimum of e is selected, which consists of α_i and β_i, and the relative location of the domain block D_k to the range block r_i. As a result of the mapping process, approximation error is calculated as follow,

$$e = d(I, W(I)) = d(I, \bigcup_{i=1}^{N_r} w_i(D_{k_i})) = \sum_{i=1}^{N_r} e_i \tag{10}$$

2.2 Decoding Scheme

At the decoder, an initial image A is prepared and divided into range blocks r_i. For each range block r_i, a domain block is located according to the received code. It is averaged, subsampled, and followed by isometrical transformation, contrast scaling and offset operations. After one mapping process, we get a transformed image A_i,

$$A_1 = W(A) = \bigcup_{i=1}^{n_r} w_i(D_{k_i}) \tag{11}$$

This mapping process is iterated until the transformed image A_n becomes same as the image A_{n-1},

$$A_2 = W(A_1) \tag{12}$$

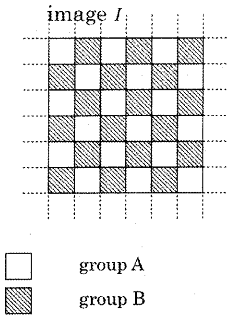
$$A_3 = W(A_2) \tag{13}$$

$$\vdots$$

$$A_n = W(A_{n-1}) \tag{14}$$

But this iteration has a limitation, which can be checked with following condition,

$$B = \lim_{i \to \infty}(A_i) \tag{15}$$



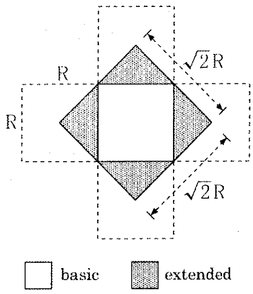


Figure 1. Groups of range blocks. Figure 2. Extension of one range block.

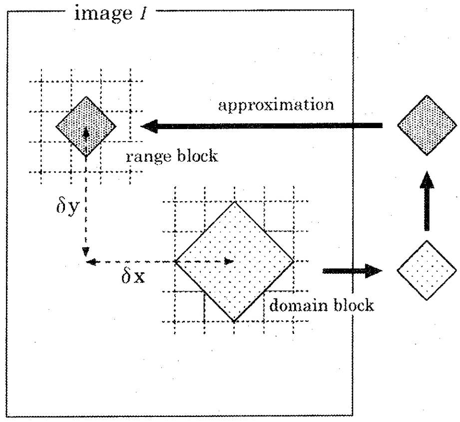


Figure 3. Encoding process.

always exists and can be written as

$$B = W(B) \tag{16}$$

3. PROPOSED SCHEME

3.1 Principle of the Proposed Scheme

In this paper, we proposed new robust fractal coding scheme. This scheme is based on the feature that the relation between the range block and domain block can be applied well to the extended range block and domain block. In this scheme, we assume that if a range block has an error, the four neighbors of that block should not have errors at that time. Based on this assumption we classify the range blocks into two complementary groups. We encode the range blocks of each group by fractal coding scheme and obtain IFS parameters like conventional scheme. We can reconstruct two whole images by extending the range blocks of each group and applying these IFS parameters to the extended range blocks. Each whole image has two kinds of portions, basic portions and extended portions. We make a basic image by collecting the basic portions of both images and an extended image by collecting the extended portions. By comparing these two images, we detect the blocks that have errors. In the actual decoding process, we make the basic and extended images in each step of IFS and replace the block with error in basic image with the same block in extended image. This prevents both the degradation of the block with errors and the undesirable propagation of error to other portions of the image.
Details of encoding and decoding processes are shown in the following.

3.2 Encoding Scheme

At encoding process we divide an image I into blocks (range block) r_i with size of $R \times R$, and classify the range blocks into two groups as shown in Figure 1. In this scheme, we need to reconstruct two kinds of whole images. Therefore we extend each range block so that each group of the extended range blocks covers the whole image. That is

$$\bigcup_{i=1}^{N_r} r_i = 2I \tag{17}$$

where N_r is the total number of range blocks.
Figure 2 shows the way to extend each range block. Each range block consists of the basic portion and the extended portion and the size of the range block is $\sqrt{2}R \times \sqrt{2}R$ as shown in Figure 2.
For each range block r_i, we search for domain block D_k, of size of $2\sqrt{2}R \times 2\sqrt{2}R$ pixels in the image I.
During the mapping process we perform the same process to domain blocks as that of conventional scheme, low pass filtering and subsampling from size $2\sqrt{2}R \times 2\sqrt{2}R$ down to $\sqrt{2}R \times \sqrt{2}R$ (same as the size of r_i) and isometrical transform, contrast scaling and offset operations.
The quality of the image constructed by the fractal image coding is related to the size of range blocks. So the quality of the reconstructed image becomes low if we make the size of range blocks larger. We carry a weight for basic portion more strongly than for extended portion. By using this weight, we prevent the degradation by making the size of range block larger.
Thus when a set of α_i and β_i is obtained, we carry a weight for RMSE (root mean squared error) e_i. In this scheme RMSE e_i is calculated by

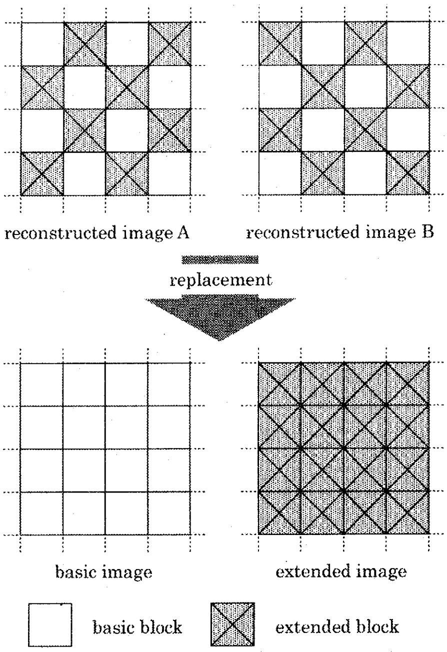


Figure 4. Replacement of blocks.

$$e_i = \sqrt{\frac{1}{R^2}\sum w_i \left\| r_i - d'_{k_i} \right\|^2} \tag{18}$$

where w_i is the weight for the extended range blocks.
By calculating e_i for N_p of domain blocks around the range block, where N_p is the number of searching domain blocks, a code for the range block which achieves the minimum of e is selected. This code consists of α_i, β_i and the relative location $(\delta x, \delta y)$ of the domain block D_{k_i} to the range block r_i like conventional scheme.

3.3 Decoding Scheme

Before decoding process, we have to preprocess to detect errors. In predecoding process, we reconstruct two whole images using the codes for each group like conventional decoding process. Both of two images consist two kind of portions, the basic portions and the extended portions. After we reconstruct two images, we construct a basic image by collecting the basic portions from two images and an extended image by collecting the extended portions from two images, which has been shown in Figure 4.
In next step, we compare the basic image with the extended image for each range blocks. In each range block, extended block consists of four parts. For each of these four parts, e is calculated by

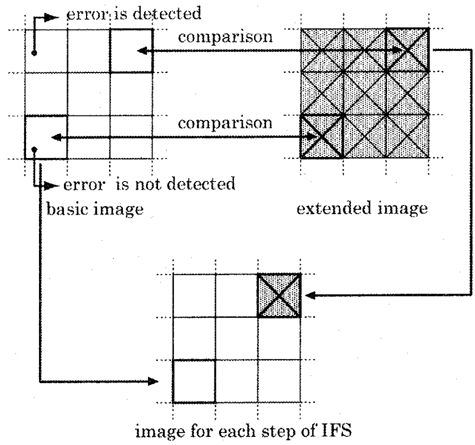


Figure 5. Example of image for each IFS Step.

$$e = \sqrt{\frac{4}{R^2}\sum \left\| B_i - E_i \right\|^2} \tag{19}$$

where B_i is a pixel of the basic image and E_i is a pixel of the extended image. If e is over threshold, we judge the error has occurred in basic part or extended part. For a same size as the basic block, the extended block consists of four extended parts. So, when the following unequality holds for a block, we judge that errors has occurred in the block of the basic image.

$$n_f > n_t \tag{20}$$

where n_f is the number of parts where we judge that error has occurred, and n_t is the number of parts where we judge no error has occurred. The information of the block where we detect errors is saved to use in decoding process.
In fractal image coding, if the code for a block has errors, the errors spread out to blocks referring to the block with errors.
The errors are also low-pass filtered, subsampled, and processed by isometrical transforms. Thus the amount of occurred errors by referring to the block with error, which we define it second error, is

$$e_2 = \frac{\alpha_2}{N^2} e_1 \tag{21}$$

where e_1 is the amount of errors, which has occurred in the referred block and α_2 is the scaling parameter of referring block and N is a ratio of a domain block side to a range block one. In the same way the amount of third, fourth, … kth error is

$$e_3 = \frac{\alpha_3}{N^2} e_2 = \frac{\alpha_3 \alpha_2}{N^4} e_2 \tag{22}$$

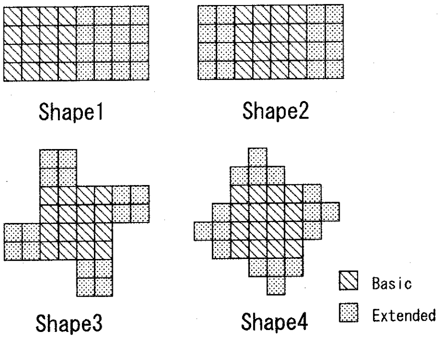


Figure 6. Shapes of extended range blocks.

$$e_4 = \frac{\alpha_4}{N^2} e_3 = \frac{\alpha_4 \alpha_3}{N^4} e_2 = \frac{\alpha_4 \alpha_3 \alpha_2}{N^6} e_1 \qquad (23)$$

$$\vdots$$

$$e_k = \frac{\alpha_k}{N^2} e_{k-1} = \cdots = \frac{\alpha_k \alpha_{k-1} \alpha_{k-2} \cdots \alpha_2}{N^{2(k-1)}} e_1 \qquad (24)$$

To prevent this propagation of errors, we perform the decoding process from beginning with the above information.

In actual decoding process, the image is reconstructed by using the block of the basic image where we don't detect errors and the extended image, where we detect errors for each IFS step. The decoding process is shown in Figure 5.

4. EXPERIMENTS

We investigate the error tolerance of the proposed and conventional scheme by adding random bit error for bits of fractal image coding data. We show a typical example of experimental results that has been done on Lena with 256×256 pixels and 8 bits grayscale. In both schemes, we don't classify range blocks into such as SHADE and EDGE, and use 4 bits for scaling parameter α, 8 bits for scaling shift parameter β, 8 bits for location information and 3bits for rotation information. The basic size of range block is 4×4.

4.1 Shape of Extended Part

The shape of extended range block is to have a same area as the basic one has. But it is free in the shape if only we can get a whole image by extended range blocks. So we investigate the SNR by changing the shape of extended range block in case of no bit errors and random bit errors. The shapes of extended range blocks are shown in Figure 6.

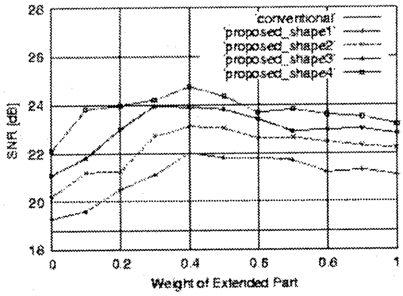


Figure 7. SNR vs. weight for extended range block (error bit rate 0.5%).

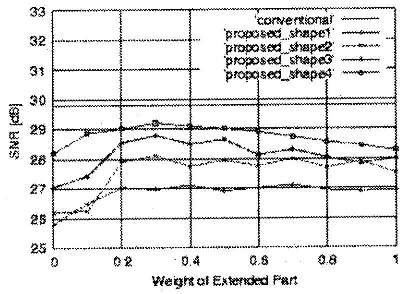


Figure 8. SNR vs. weight for extended range block (error bit rate = 0%).

Figure 7 and 8 show the SNR vs. weight for the extended range block by changing the shape of extended range block. About the weight for the extended range block, we mention in the next section. These figures suggest that the Shape4 of the extended range block scores the best result in SNR with any weights for the extended range block. We think that is because it is smallest that the mean distance between the pixel of the extended range block and basic range block so that we can apply Collage theorem, that is the basic theory in Fractal, most easily to not only the basic range portion but also the extended portion in the shape4.

4.2 Weight for Extended Range Block

In the Figure 7 and 8, the horizontal axis is the w_i in equation (18). The weight for the basic portion is also 1.0 and the threshold shown in equation (19) is 15 in this experiment.

At random error bit rate of 0%, the SNR for $w_i = 0.0$ is less than when $w_i = \sim 0.3$ because the amount of wrong error detections is increased. On the other hand, wrong error detections of high weight is less than low weight but the SNR of high weight is not better than low weight because the degradation of image quality by extending range blocks is more than the degradation by wrong error detection. In random bit error rate of 0.5% the SNR of $w_i = 0.4$ scores the best result of all because of the same reason.

4.3 Result after Optimization

Next we optimize the shape of the extended range block and the weight for extended range block; according to the result of the previous two sections, we fix the shape of extended range block as shape4 and weight for the extended range block as 0.4. Figure 9 shows the SNR vs. bit error rate. The proposed scheme scores 2-3 dB better than the conventional one in case of bit errors and scores nearly to the conventional one in case of no errors.

The reconstructed images of both schemes are shown at the random bit error rate of 0% and 0.5% in Figure 10 and 11. In case of 0.5% errors, we can see the most big errors are concealed in the proposed method and the difference between two method is clear. In case of 0% errors, we can hardly see the difference between two pictures because the SNR difference is little, and still more, the big errors by wrong error detection do not occur in proposed scheme.

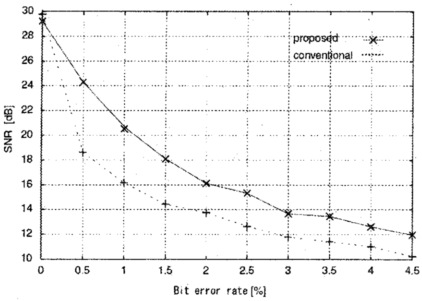


Figure 9. SNR vs. bit error rate.

(a) Conventional 18.7dB (b) Proposed 24.2dB

Figure 10. Decoded images (error bit rate = 0.5%).

(a) Conventional 29.9dB (b) Proposed 29.1dB

Figure 11. Decoded images (error bit rate = 0%).

5. CONCLUSION

We proposed a new fractal-coding scheme that can detect and correct errors, without increasing the amount of the data. It has nearly the same performance as that of the conventional scheme when error doesn't occur and achieves large improvement of image quality when errors occur.

REFERENCES

[1] Ito, S. Usui, T. Fujii, T. Kimoto, M. Tanimoto : "Error Resilience in Fractal Image Coding," Tokai-Section Joint Conference. 542. p. 271 (September 2000)

[2] M. F. Barnsley, FRACTALS EVERYWHERE, Academic Press Professional, New York, 1993.

[3] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformation," IEEE Trans. Image processing 1, pp. 18-30, 1992

[4] M. Harada, T. Kimoto, T. Fujii, M. Tanimoto : "Fast Calculation of IFS Parameters for Fractal Image Coding," PCIP 2000 1, pp. 457-464