FAST FRACTAL IMAGE ENCODER DESIGN

Yung-Gi Wu,
Department of Computer Science and Information Engineering Institute of Applied Information, Leader University, Tainan, Taiwan
wyg@mail.leader.edu.tw

DOI: 10.36724/2664-066X-2021-7-4-40-44

ABSTRACT

Fractal theory has been widely applied in the filed of image compression due to the advantage of resolution independence, fast decoding, and high compression ratio. However, it has a fatal shortcoming of intolerant encoding time because that every range block is need to find its corresponding best matched domain block in the full image. Therefore, it has not been widely applied as other coding schemes in the field of image compression. In this paper, an algorithm is proposed to improve this time-consuming encoding drawback by the adaptive searching window, partial distortion elimination and characteristic exclusion algorithms. Proposed can efficient decrease the encoding time significantly. In addition, the compression ratio is also raised due to the reduced searching window. Conventional fractal encoding for a 512 by 512 image need search 247009 domain blocks for every range block. Experimental results show that our proposed method only search 120 domain blocks which is only 0.04858% compared to conventional fractal encoder for every range block to encode Lena 512 by 512 8-bit gray image at the bit rate of 0.2706 bits per pixel (bpp) while maintaining almost the same decoded quality as conventional fractal encoder does. This paper contributes to the research of encoder of fast image communication system.

KEYWORDS: *Fast communication, fractal encoder, compression.*

The article is reworked from unpublished 2nd IEEE International Conference on Circuits and Systems for Communications (ICCSC) materials.

1. INTRODUCTION

A picture may be worth a thousand words, but it requires far more memory to store or bandwidth to transmit. With the successes of multimedia technology and the era of wideband network, peoples' desire to the high quality of multimedia still can not be satisfied. All the scholars in the universities and the engineers in the industry want to get the compromise between the limited network bandwidth and the unlimited human desire. Among all the digitized data that we people can touch every day, such as digital library, VCD, DVD, JPEG, etc, the kernel of the system or the standard that commercial products use is the compression technique.

There are many coding schemes that have been developed such as DCT, VQ, Wavelets, BTC, Fractal, etc. In general, the criteria to evaluate the performance of a compression system include 1) compression ratio 2) reconstructed quality 3) processing time. Fractal compression can achieve the highest compression ratio among all the existed coding schemes theoretically; however, its encoding time is terrible intolerant to practical industrial applications. If this shortcoming of long encoding time can be improved, the application of fractal will be a practical consideration.

The cause of fractal image coding with high compression is that the minority of blocks through rotations represent the majority of blocks. In a word, fractal encoding is based on Partitioned Iterated

Function System (PIFS). The detailed descriptions of PIFS can be found in [2], [3], [4]. To improve the drawback of time consuming in encoding process, we utilize classification and local search techniques to exclude those un-related searching domains to reduce the encoding time while decreasing the bit rate as well in this paper. The remaining sections are organized as follows: Section 2 will depict the basic fractal image coding. Proposed method is given in Section 3 and results will be shown in Section 4.

FRACTAL IMAGE CODING

Fractal image coding is based on partition iterated function system (PIFS). Let an original image be partitioned into non-overlapping regions called range blocks (R) and overlapping regions called domains blocks (*D*). The size of each domain block should be larger than that of the range block to satisfy the property of contraction. Let *D*' denotes the down sampled domain block of *D* and the *D*' size is equal to range blocks to match the contractive property.

The transformations are composed of a geometric transformation and a massic transformation. The geometric transformation consists of moving the domain block to the location of the range block and adjusting the size of domain block to match the size to size range block. The massic transformation adjusts the intensity and orientation of the pixels in the domain block after it has been operated on by the geometric transform. The massic transformation t_i can be depicted as follows:

For each range block, we must find the best matching domain block (*D*') by the affine transformations

$$t_i\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (1)$$

where si controls the contrast and oi controls the brightness. $Z = f(x, y)$ is the gray level value at (x, y) and a_i, b_i, c_i, d_i, e_i, f_i denote the eight-symmetry such as (1) Identity mapping (2) Rotation by 90 degrees (3) Rotation by 180 degrees (4) Rotation through -90 degrees (5) Reflection about mid-vertical axis (6) Reflection about mid-horizontal axis (7) Reflection about diagonal (8) Reflection about cross diagonal. The i in s_i and o_i denotes one of the above mentioned symmetries which means $(1<= i <=8)$.

In practice, we compare a range block and down sampled domain blocks using RMS metric as follows

$$RMS = \sum_{k=1}^{n\times n}\left(s_i \times a_k + o_i - b_k\right)^2 \quad (2)$$

Where ak represents the pixel value of the domain blocks (*D*') after eight transformations and b_k represents the pixel value of the range blocks and the block size for both *R* and *D*' is n by *n*. This RMS metric allows easy computation for optimal values of s_i and o_i in equation (1). This will give us contrast and brightness settings that make the affinely transformed a_k values have the least squared distance from the b_k values. The minimum of RMS occurs when the partial derivatives with respect to s_i and o_i are zero, which occurs when

$$si = \frac{n\times n\left(\sum_{k=1}^{n\times n} a_k b_k\right) - \left(\sum_{k=1}^{n\times n} a_k\right)\left(\sum_{k=1}^{n\times n} b_k\right)}{n\times n\sum_{k=1}^{n\times n} a_k^2 - \left(\sum_{k=1}^{n\times n} a_k\right)^2} \quad (3)$$

$$oi = \frac{\sum_{k=1}^{n\times n} b_k - si\sum_{k=1}^{n\times n} a_k}{n\times n} \quad (4)$$

There are many best matched criteria to choose. The root mean square (RMS) is usually used in fractal image coding and the minimal RMS is the better matching. We use equation (2) to find the optimal si and oi and then quantize them for storage or transmission.

In addition, the encoder must record the position of the best matched domain block (*D*') and its transformation for each range block so as to reconstruct the decoded block on the decoder side.

The following example depicts how this encoding can be done. Suppose the data to be dealt with is 512 x512 pixel image in which each pixel can be one of the 256 levels of gray(ranging from black to white). Let Ri be the 8x8 pixel non-overlapping range block (i=1,..,4096) and let D be the collection of all the 16x16 overlapped sub-squares of the image.

The collection of D contains 497x497=247009 squares. For each R_i, search through all of collection of D_i to find one which minimizes the RMS as equation (2); that is, find the part of the image that most looks like the image above R.

There are 8 ways to map one square onto another, so that this means comparing 8x247009=1976072 squares with each of the 4096 range blocks. In addition, we must fulfill down-sampling for each Di to get the same size of R to satisfy the property of contraction.

Choosing 1 from each 2x2 sub-square of D_i or averaging the 2x2 sub-square corresponding to each pixel of R can achieve the goal of down-sampling. From the above descriptions about the conventional fractal encoding, we know the huge computation overhead is obviously. The time to search the best matched domain block for every range block is intolerant a time consuming job in practical application.

Therefore, we develop a new encoding algorithm to reduce the time in this research. In addition, the bit rate is also be reduced by the smaller search window. A lot of people make efforts in fractal improvement. Some investigate region-based image coding methods in [5] and some combine fractal with other algorithm such as wavelet in [6], genetic algorithms in [7], discrete cosine transform in [8], [9]. In a word, they make use of classification methods to reduce time.

FAST ENCODING ALGORITHM

The major factor to cause the huge computation for fractal encoding is the searching number of domain blocks. Therefore, decreasing the number of it is an institute method to speed up the encoding time. The methods to achieve the goal in proposed method are by classifying range and domain blocks, respectively and searching the algebraic adjacent region for every range block. Following sub-sections depict the detailed descriptions of proposed method.

The classification approach has the advantage that it elegantly excludes the range block to search those domain blocks whose characteristics do not match the characteristic of range block.

Here, variance of a block is used to be the basic criterion of classification. The formulas to get the variance of block X are given as follows:

$$u(X) = \frac{1}{n \times n} \sum_{i=1}^{n \times n} X_i \tag{5}$$

$$\mathrm{var}(X) = \frac{1}{n \times n - 1} \sqrt{\sum_{i=1}^{n \times n} (X_i - u(X))^2} \tag{6}$$

nxn is the block size of X. There are four classes of c1, c2, c3 and c4 after the classification. c_1 is classified by variance merely once the condition var(X)<=T1 is met. Those range blocks belonged to c1 class are encoded and transmitted or storage by the mean value $u(X)$ only. The other range blocks whose classifications are not c1 will be encoded by fractal and those blocks are classified into c_2, c_3 and c_4. The criterion to classify those blocks into (c_2, $c_3$3, c_4) is according to the distortion after fractal encoding. Detailed algorithm is depicted as following:

Input: range blocks (R) whose var(R)>T1

Output: c_i ($i \in \{1,2,3\}$)

Step 1: Implement fractal encoding to find the best matched D within the searching window and record its distortion RMS by equation (2).

Step 2: If RMS>T2, output c_4;

else ((RMS>T3)&&(RMS<=T2)) output c3;

else output c_2;

We set three thresholds {T1, T2, T3} to get the classification.

Note that T1 is the variance of input range block for classification and {T2, T3} is the RMS distortion to classification. Refer to Figure 1 and Figure 2, which are the resulted regions after classification when the threshold set is to {0.5, 25000, 15000}.

Figure 1 is c1 class regions.

Fig. 1. Pure blocks after classification

There are three classes regions shown in Figure 2.

Fig. 2. Fractal encoding regions

The regions marked by light lattice denote the c_4 region, dark lattices represents c_3 region and the other regions without lattice belong to c_2. Here, the size of each range block is 8 by 8. c_1 blocks are compressed or transmitted by the mean value of itself and c_2 ~c_4 blocks are compressed or transmitted by fractal encoding. The following sections depict the reduction of searching window and the computation reduction method to find the best matched domain block for each c_2 ~ c_4 range blocks.

If the size of each overlapped domain block (*D*) is 16x16 and the we move *D* pixel by pixel to cover the whole image, there are 247009 domain blocks for a 512 x 512 image. A range block is encoded by projecting it tentatively on to the entire 247009 domain blocks in conventional fractal encoding. Therefore, its computation burden is terrible huge. Proposed method decreases the searched number of domain blocks to about 120 while losing a little fidelity. For each range blocks in c_2 ~ c_4 classes, the searching window (SW) is only constrained to the nearby regions instead of the entire domain pool and we do not move *D* pixel by pixel within the SW to reduce the computation burden furthermore.

The step size to move *D* within the searched window is set to 4 pixels in the later experiments. In addition, we exclude the domain blocks whose variances do not match the range block to be encoded within search window. Detail descriptions of the reduction of searching window for c_2 ~ c_4 range blocks are given as follows:

Input: Class of range block (*cx*) and position of range block (*i,j*), image_size=MxN.

Output: range of search window.

```
        switch (cx)
{
        case (c2):  SW=SW_2;  break;
        case (c3):  SW=SW_3; break;
        case (c4):  SW=SW_4; break;
}
search_range_of_i= (i-SW) ~ (i+SW);
search_range_of_j= (j-SW) ~ (j+SW);
if (i-SW)<0  search_range_of_i=0 ~ 2 x SW;
if(i+SW)>M search_range_of _i=(M-2 x SW) ~ M;
if (j-SW)<0  search_range_of_j=0 ~ 2 x SW;
if(j+SW)>N search_range_of _j=(N-2 x SW) ~ N;
```

If the size of the domain block is 16 by 16 and we move the domain block pixel by pixel then the total searched number of domain block is *(2xSW-15)x(2xSW-15)*. In this research, we move the range block four pixels every time so that the number of the searched domain block is

$$\frac{2\times SW-15}{4}\times\frac{2\times SW-15}{4} \tag{7}$$

Compared to the conventional fractal encoding algorithm whose searched numbers of domain block for every range block is (M-15)x(N-15) when the image size is *M* by *N* and the size of the domain block is 16 by 16, the economize searched number of domain block is given in (8)

$$\frac{\frac{(2\times SW-15)^2}{16}}{(M-15)\times(N-15)} \tag{8}$$

Let the image size be 512 by 512 and the SW be 32, it only searches about 144 domain blocks nearby the range block in the proposed method, which is only about 1/1715 compared to the conventional fractal encoding scheme. After selecting the searching window for every range block, the next step is to calculate as equation (2) to find the best matched one and its correspondent transformation within the window. Note that for every input range block whose class is not c1, we treat it as c2 class at first. If it meets the criterion of classification as described in the previous section, it will select larger searching window size to find better domain block.

The major factor to lead the fractal encoding become a time consuming technique is the vast computation cost on RMS. Previous section depicts the algorithm to select the nearby domain blocks which decreasing the number of RMS of computation instead of the whole image. This section uses another methodology to decrease the computation furthermore.

For every selected domain block, we must compute the error between the range block and the eight-transformed domain blocks to find the best matched one whose error is minimum. We use a method to reduce the computation of RMS metric. This algorithm is called as Partial Distance Elimination (PDE) which is devised by Bei and Gray in 1985 for the application of fast Vector Quantization encoding. The PDE algorithm discards all domain blocks whose partial distortion relative to an input vector exceeds the current available nearest distortion. The operation of the PDE algorithm is summarized as follows:

Input: Range block (R);

Selected sampled domain blocks D' (number is z);

Output: Best matched D' and its best isometric;

Step1: find si and oi by equation (3)(4) for first D';

i=0;

$$RMS=\sum_{k=1}^{n}\sum_{l=1}^{n}(D_1^{'}(k,l)\times s_i+o_i-R(k,l))^2 \tag{9}$$

```
        current_error =RMS;
        best_D'=1;
        best_isometric=0;
Step 2:
    for(p=1;p<=z; p++)  {
    find si and oi by equation (3)(4) for every  D' ;
    if(p==1)  a=1; else a=0;
// the RMS of first D' and its first isometric has calculated already
in Step 1;
            for(i=a; i<8;i++) {
            RMS=0;
            for(k=1; k<=n; k++)
             for(l=1; l<=n; l++)
              {
```

$$RMS+=(D_i'(k,l)\times s_i+o_i-R(k,l))^2 \quad (10)$$

```
    if (RMS>=current_error)  goto exit;
  }
If(RMS<current_error)
{
current_error=RMS;
best_D'=p;
best_isometric=m;
}
exit: { };
    }
}
return (best_D', best_isometric);
```

There are z D_i' (i=1..z) to be compared to R and every D' has eight-symmetry forms; thus, the total RMS computation is zx8. Step 1 calculates a RMS error between R and the first isometric of D_1'. Step 2 calculates all the other RMS including the other seven isometrics of D_1' and the eight isometrics of all the D_2' ~ D_z'; meanwhile, it compares the distortion between the current minimum error and increasing RMS in (10). As long as the increasing RMS exceeds the current minimum error, it stops the RMS calculation and exits to the outer of the loop. Such a method can decrease the heavy computation of RMS effectively.

SIMULATION RESULTS AND CONCLUSION

In order to evaluate the performance of the proposed method, several images including Lenna, Lena are employed to test. of the test images are 512 by 512 with 8-bit gray level. The circumstance of our experiment was on a single PC with Pentium 4-2.4GHz CPU and 256 MB RAM. In our discussion of the results, we will focus on the visual quality, encoding time, compression ratio, and PSNR. Visual quality is an objective judgment and PSNR is an evaluation in mathematic sense.

The bit rate for a 512 by 512 image whose size for a range block is 8 by 8 is calculated by the following:

$$BR=\frac{\#c1\times 8+(\#c2\times(2\times sw\_c2-15)^2+\#c3\times(2\times sw\_c3-15)^2+\#c4\times(2\times sw\_c4-15)^2)/16+(\#c2+\#c3+\#c3)\times(15)}{512\times 512}+\frac{2}{64} \text{ bpp} \quad (11)$$

#*cn* denotes the number of each class. *sw_cn* represents the searching window size of class *n*. Because the c1 class range blocks are all pure, only the mean value of each one is transmitted instead of the fractal encoding.

The second term (#c2× (2× sw _ c2 −15)2 +#c3× (2× sw _ c3 −15)2 +#c4× (2× sw _ c4 −15)2) /is the bits used to represent position of the searched D within the searching window for c2 ~ c4 classes. The size of each D is 16 by 16 and moves the D four pixels a time. (#c2+#c3+#c4)×15 is used to specify the 8 transformations (3-bit) and si (7-bit) and oi (5-bit) for each R in class c2~c4. The last term 2/64 is used for classification of each range block.

The resulted data of encoding time, searched domain blocks and decoded quality after running to all the four images by the proposed method and conventional fractal encoding are given in table 1.

Table 1

Simulation Results

	Lena	L
Bit_rate(bpp)	2706	0.2475
Compression Ratio	29.562	32.322
Encoding time (second)	15.89	12.546
PSNR(dB)	28.9641	30.0904

The thresholds of classification (T1, T2, T3) is (0.5, 25000, 15000) and the searching window size for class_2, class_3, class_3 is 8, 16 and 64, respectively. The table lists the number of each class. The blocks belongs to Class_1 is pure so that fractal encoding is not employed.

The average searched domain blocks for each range block with respect to Lena and Lenna are 129 and 120, respectively. Practical running time is 15.89 and 12.546 seconds. Due to the reduced search window for each range block, the bits used to the represent best matched domain block also decreased.

Refer to table 1, the bit rate is 0.515625 bpp for conventional fractal encoding and the proposed method decreases the bit rate to 0.2706 bpp and 0.2475 bpp for Lena and Lenna, respectively. Running time is 28685 and 28681 seconds for the two test images and bit rate for the two images are 0.515625 bpp by using the conventional fractal encoding. Because the numbers of searched domain blocks are restricted to the searching window, the decay of quality is indeed irreproachable. However, the visual difference is almost unnoticeable.

REFERENCES

[1] M. F. Barnsley. *Fractal everywhere*, Academic Press, New York, 1988.

[2] Y. Fisher. *Fractal Image Compression. Theory and Application*, New York: Springer-Verlag, 1994.

[3] A.E. Jacquin. Fractal image coding: a review, *Proceedings of the IEEE*, Vol.81, No.10, pp.1451-1456 Oct.1993.

[4] A.E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Trans.Image Processing*, vol. 1, pp.18-30, Jan. 1992.

[5] B. Wohlberg and G. de Jager. A review of the fractal image coding literature, *IEEE Trans. Image Processing*, vol. 8, pp. 1716-1729, Dec. 1999.

[6] G.M. Davis, A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Processing*, vol. 7, pp. 141-154, Feb. 1998.

[7] M. Takezawa, H. Honda, J. Miura, H. Haseyama and H. Kitajima. A genetic-algorithm based quantization method for fractal image coding, *IEEE Trans. Image Processing*, vol. 1, pp. 458-461, 1999.

[8] Y. Zhao and B. Yuan, "Image compression using fractals and discrete cosine transform," *Electron. Lett.*, vol. 30, no. 6, pp. 474-475, 1994.

[9] Y. Zhao and B. Yuan. A hybrid image compression scheme combining block-based fractal coding and DCT, *Image Communication.* Vol. 8, No. 2, pp. 73-78, Mar. 1996.

[10] C. Bei and R.M. Gray. An improvement of the minimum distortion encoding algorithm for vector quantization, *IEEE Trans. Commun*., vol. COM-33, pp. 1132 1133, Oct. 1985.