Fractal Image Compression with Error Control

Shou-Cheng Hsiung
Dept. of Information Engineering
I-Shou University
Kaohsiung County, Taiwan
886-7-6577251
d9303004@stmail.isu.edu.tw

J. H. Jeng
Dept. of Information Engineering
I-Shou University
Kaohsiung County, Taiwan
886-7-6577251
jjeng@isu.edu.tw

D. J. Duh
Dept. of Computer Science and Information Engineering
Ching_Yun University, Taiwan
886-3-2503013#7701
duby@saturn.yzu.edu.tw

ABSTRACT
Fractal image compression is of great interest both in theory and application. However, applications of fractal-based coding to other aspects of image processing have received little attention. In this paper, we propose a fast encoding algorithm to enhance and restore a noisy image, and utilize the coding scheme for transmission through the wireless fading channel. By using the predicted frequency-like contrast coefficients, we simplify the calculations for the contrast and brightness parameters, and suppress a significant amount of the noise. Experimental results show that the encoding time is about two times faster than that of the full search method, and also gain better quality for the deblurred image.

Categories and Subject Descriptors
I.4.3 [**Image Processing and Computer Vision**]: Enhancement – *Sharpening and deblurring*

General Terms
Algorithms

Keywords
Fractal Image coding, image denoising, image restoration

1. INTRODUCTION
The need for image restoration is encountered in many practical applications. Image distortion can be caused by poor quality image acquisition, images observed in a noisy environment or noise inherent in communication channels. There exist various adaptive and nonlinear image restoration methods that account for the variations in the local statistical characteristics [1], [2]. In this paper, we explore an method of image restoration based on fractal image coding.

The idea of the Fractal image compression is based on the assumption that the image redundancies can be efficiently exploited by means of block self-affine transformations. Many authors [3], [4] suggest that at a block level images contain a large amount of self-similarity, and that the Fractal transform coding can be used to take advantage of this fact. The Fractal transform for image-data compression was introduced first by M. F. Barnsley and S. Demko [5]. However, the algorithm requires extensive computations [6]. The first practical Fractal image compression scheme was introduced by A. E. Jacquin [7], E. W. Jacobs, R. D. Boss and Y. Fisher [8] called the Jacquin-Fisher algorithm using block-based transformations and an exhaustive search strategy. Their approach was an improved version of the system patented by Barnsley [9]. The computational complexity was reduced by partitioning and classifying the image sub-blocks.

There are two important issues which need to be improved in the Jacquin-Fisher algorithm. First, the cost of an exhaustive search of a pool of domain blocks is too high. Secondly, a good classification algorithm needs to be obtained. Such a classification algorithm must be able to reduce the search space and also to decrease the order of the fractal function. There is also a need to consider the eight orientations of the allocated domain blocks. These are known as the dihedral operations on the block. This group of eight operations involve extensive computations in order to realize the above encoding procedure. It is well known [10], [11], [12] that a fast transform method can be used to implement the DCT, called the fast DCT. By performing the MSE computations in the frequency domain, after the proper arrangement, it follows that all of the redundant computations can be eliminated.

However, little attention has been given to the use of such fractal-based methods for the purpose of image enhancement and restoration. M. Ghazel etc. show that straightforward fractal-based coding performs rather well as a denoiser [13], [14]. Since most of the energy of a block is located in the low-band region, we develop a frequency-like transformation in spatial domain to acquire only three coefficients to represent the block. This reduces the complexity of the inner product computations. Also, the eight operations of the Dihedral group can be directly obtained by changing the signs of the three frequency-like coefficients. Therefore, the purpose of both speedup and denoising can be achieved.

2. FRACTAL CODING
Fractal image compression is an inverse problem, i.e., for the given set S_W, find the IFS which has S_W as its attractor. It should be noted that, when S_W is a natural image, such IFS can hardly exist because most of the sub-images are not directly similar to the whole image. To solve this problem, the idea of local self-similarity is adopted to form the Partitioned Iterated Function

Mobimedia 2008 July 7-9, 2008, Oulu, Finland.

DOI 10.4108/ICST.MOBIMEDIA2008. 4041

System (PIFS) in which the contractive maps w_i is defined only on D_i where $D_i \subset X$ for $i = 1,...,n$.

For practical implementation, let f be a given 256×256 gray level image. The domain pool D is defined as the set of all possible blocks of size 16×16 of the image f , which makes up $(256-16+1)\times(256-16+1)$ $=58081$ blocks. The range pool R is defined to be the set of all non-overlapping blocks of size 8×8 , which makes up $(256/8)\times(256/8)=1024$ blocks. For each block v from the range pool, the fractal transformation is constructed by searching in the domain pool D the most similar block. At each search entry, the domain block is sub-sampled such that it has the same size as the range block. Then, the sub-sampled block is transformed subject to the eight transformations in the Dihedral group on the pixel positions. Assuming the origin of the image block is located at the center, the eight transformations $T_k : k = 0,\ldots,7$ can be represented by the following matrices:

$$T_0=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\quad T_1=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix},\quad T_2=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix},\quad T_3=\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}$$

$$T_4=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix},\quad T_5=\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix},\quad T_6=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix},\quad T_7=\begin{bmatrix}0 & -1\\ -1 & 0\end{bmatrix}$$

The transformations T_1 and T_2 correspond to the flips along the vertical and horizontal lines, respectively. T_3 is the flip along both the vertical and horizontal lines. T_4 , T_5 , T_6 and T_7 are the transformations of T_0 , T_1 , T_2 and T_3 , respectively, performed by an additional flip along the main diagonal line.

Let u denote a sub-sampled domain block of the same size as v . The similarity of u and v is measured using Mean Square Error (MSE) defined by

$$d(u,v)=\frac{1}{64}\sum_{j=0}^{7}\sum_{i=0}^{7}(u(i,j)-v(i,j))^2 \tag{1}$$

In another word, for each range block R_i , the fractal encoder searches from the set of domain blocks a domain block D_i and a contractive mapping w_i which minimizes $d(R_i, w_i(D_i))$ where

$$w_i\begin{bmatrix}x\\ y\\ z\end{bmatrix}=\begin{bmatrix}a & b & 0\\ c & d & 0\\ 0 & 0 & p\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}+\begin{bmatrix}t_x\\ t_y\\ q\end{bmatrix}. \tag{2}$$

The transformations w_i are also called the affine transformation. The parameters (*a, b, c, d*) constitute the eight Dihedral transformations, (t_x , t_y) is the position of the domain block, p is the contrast scale factor and q is the luminance offset. In the minimization process, p and q can be computed directly as

$$p_k=\frac{\left[N<u_k,v>-<u_k,\vec{1}><v,\vec{1}>\right]}{\left[N<u_k,u_k>-<u_k,\vec{1}>^2\right]} \tag{3}$$

and

$$q_k=\frac{1}{N}\left[<v\cdot\vec{1}>-p<u_k\cdot\vec{1}>\right] \tag{4}$$

where $N = 64$ and $\vec{1}=\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^T$.

After the appropriate affine transformations was found for all range blocks, the parameters t_x , t_y , p , q and the orientation of eight Dihedral transformations are stored, which are referred to as the fractal codes.

In the decoding phase, one chooses an arbitrary initial image, and then uses the fractal codes to compute the attractor of each transformation w_i . After appropriate times of iterations, the image can be reconstructed, which has some degree of loss corresponding to the original image.

The exhaustive search algorithm proposed by Jacquine and Fisher, the optimum values for p and q are found by the method of a MSE for all of the 8 cases of the isometry transformations for each domain block in the domain pool.

Given a range block v , one searches the domain block and the affine transformation w in the domain pool by the minimum MSE $\varepsilon^2=\left|w(u)-v\right|^2/N$, where N is the number of pixels in the block and u belongs to the domain pool. To simplify the notation, one assumes that all of the domain blocks are shrunk already in the x-y plane by the factor 1/2. Thus all the domain blocks are of the same size as the range blocks.

At each search entry u and its 8 orientations, the quantities p and q are computed as

$$p=<\hat{u}_k,\hat{v}>/<\hat{u}_k,\hat{u}_k> \qquad \textit{and} \qquad q=m_v-pm_u \tag{5}$$

where $<\hat{u}_k,\hat{v}>$ denotes the inner product of $\hat{u}_k$ and $\hat{v}$, m_u and m_v are the mean values of the blocks u and v , respectively, $\hat{u}=u-m_u$ and $\hat{v}=v-m_v$ are the demeaned blocks of u and v , respectively. Hence, the MSE can be re-formulated as

$$\begin{aligned}\varepsilon^2&=\frac{1}{N}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}((p\cdot u_k(i,j)+q)-v(i,j))^2\\&=\frac{1}{N}(<\hat{v},\hat{v}>-p<\hat{u}_k,\hat{v}>)=\frac{1}{N}(<\hat{v},\hat{v}>-\frac{<\hat{u}_k,\hat{v}>^2}{<\hat{u}_k,\hat{u}_k>})\end{aligned} \tag{6}$$

Thus, it is clear that the minimization of the MSE of u_k and v is equivalent to the maximization of the inner product $<\hat{u}_k,\hat{v}>$.

3. THE PROPOSED METHOD

In equation (3), the range block v and domain block u_k consist of 64 pixel values in spatial domain. Inspired from the fast DCT algorithm, we develop a new simple transformation to spare the cost of DCT computation. We first derived the mean values of the four sub-blocks, as shown in figure 1. By manipulating the four mean values, we can get three frequency-like coefficients.

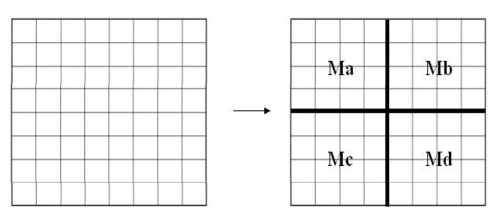

Figure 1. Mean values of the four sub-blocks.

Let b be a given 8×8 image block. The frequency-like transform of b is a 3-dim vector $\hat{b} = (\beta_1, \beta_2, \beta_3)$ given by

$$\begin{aligned} \beta_1 &= (M_a + M_c) - (M_b + M_d) \\ \beta_2 &= (M_a + M_b) - (M_c + M_d) \\ \beta_3 &= (M_a + M_d) - (M_b + M_c) \end{aligned} \tag{7}$$

The frequency-like coefficients of $\hat{b}_k$, where $b_k, k = 0..7$ are the Dihedral transformation of b, can be easily computed as

$$\begin{aligned} b_0 &= (\beta_1, \beta_2, \beta_3), & b_4 &= (\beta_2, \beta_1, \beta_3), \\ b_1 &= (-\beta_1, \beta_2, \beta_3), & b_5 &= (-\beta_2, \beta_1, \beta_3), \\ b_2 &= (\beta_1, -\beta_2, \beta_3), & b_6 &= (\beta_2, -\beta_1, \beta_3), \\ b_3 &= (-\beta_1, -\beta_2, \beta_3), & b_7 &= (-\beta_2, -\beta_1, \beta_3) \end{aligned}$$

As a consequence, there is no need to separately calculate the frequency -like coefficients of the 8 Dihedral transformations. Since these coefficients are the predicted values for the DCT low-band data. Equations (3) for the contrast parameter has changed to

$$p_k = < \hat{v}, \hat{v} > / < \hat{u}_k, \hat{v} > \tag{8}$$

where $\hat{v}$ and $\hat{u}_k$, the transformed blocks, consist of only three integers. The steps to construct the transformed blocks are given as follows.

Step1. Divide each 8×8 block into four blocks.
Step2. Compute the four mean values.
Step3. Derive the three frequency-like parameters.
Step4. Form the transformed block.

As shown previously, the Dihedral transformed domain blocks needs only the three coefficients. This reduces the complexity of inner product in equation (6). Consequently, our fast encoding algorithm requires less computations than the original algorithm.

4. EXPERIMENTAL RESULTS

To emphasize the encoding speed with a constant quality, the fast algorithm is compared to the full-search algorithm having the same compression ratio. The data is encoded using 8 bits for the translations t_x and t_y, respectively, 7 bits for the brightness q, 5 bits for the contrast p and 3 bits for the dihedral transformation. A total of 31 bits are needed for coding each range block. The PSNR given by $PSNR = 10 \cdot \log_{10}(255^2 / MSE)$ is used to measure the quality of the retrieved images. For the different algorithms, the retrieved image is obtained by the same number of iterations of the encoded affine transformation with the same initial image.
The software simulation is done using C++ Builder 5 on a Pentium 1.5G, windows XP pc. The tested pattern is the gray level Lena, Peppers and Baboon images of size 256×256 with various percentage of Pepper and Sault noise added. Table 1 gives the comparison results of encoding time of *p+q*. Table 2, 3 and 4 show the decoded image quality between the full search method and the proposed method. Figure 2 and 3 show the decoded images by using the full search method and the proposed algorithm. Experimental results show the proposed algorithm is about two times faster than that of the full search method, and gets better quality of deblurred image.

Table 1. The encoding time of *p+q* (in seconds) comparison.

	Full search method	Proposed Method
Lena	325	171
Peppers	326	175
Baboon	325	172

Table 2. The decoded image quality comparison with Lena.

Noise added	Full search method	Proposed method
0%	29.17	28.58
2%	23.03	24.01
4%	21.50	22.85
8%	20.84	21.62
15%	18.80	19.48

Table 3. The decoded image quality comparison with Peppers.

Noise added	Full search method	Proposed method
0%	29.85	29.32
2%	23.35	24.76
4%	22.57	23.85
8%	21.16	21.66
15%	19.46	20.50

Table 4. The decoded image quality comparison with Baboon.

Noise added	Full search method	Proposed method
0%	21.80	21.57
2%	20.62	20.46
4%	20.05	20.01
8%	19.44	19.46
15%	18.67	18.35

Figure 2. (a) Lena with 2% noise added (b) with 8% noise added
(c) Full search method decoded 2% noise, PSNR=23.03
(d) Full search method decoded 8% noise, PSNR=20.84
(e) Proposed method decoded 2% noise, PSNR=24.01
(f) Proposed method decoded 8% noise, PSNR=21.62

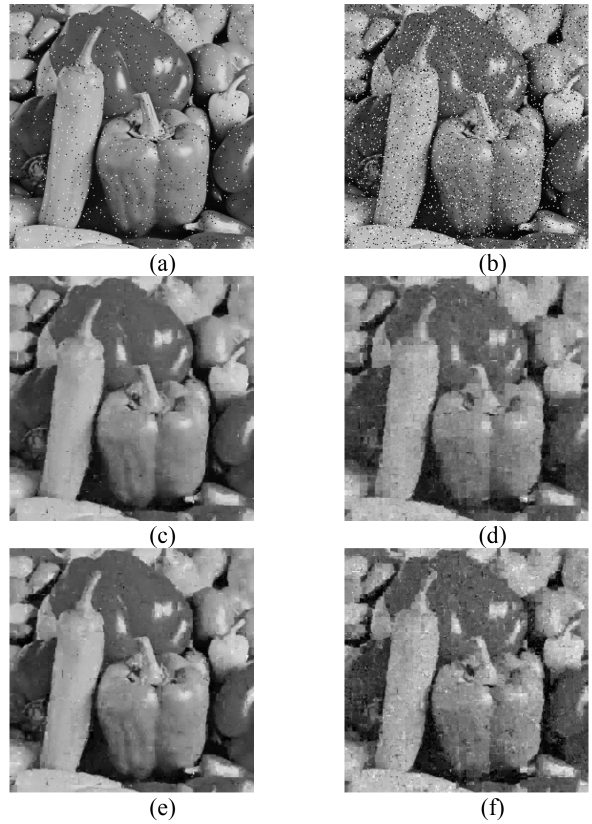

Figure 3. (a) Peppers with 2% noise added (b) with 8% noise added
(c) Full search method decoded 2% noise, PSNR=23.35
(d) Full search method decoded 8% noise, PSNR=21.16
(e) Proposed method decoded 2% noise, PSNR=24.76
(f) Proposed method decoded 8% noise, PSNR=21.66

5. REFERENCES

[1] R. Pinter, Ed., Nonlinear Vision. Boca Raton, FL: CRC, 1992.

[2] I. Pitas and A. N. Venetsanopoulos, Nonlinear Digital Filters. Norwell, MA: Kluwer, 1990

[3] H. O. Peitgen, J. M. Henriques, and L. F. Penedo, Fractals in the fundamental and applied sciences, Elsevier Science Publishing Company Inc. New York, 1991.

[4] A. J. Crilly, R. A. Earnshaw, and H. Jones, Fractals and Chaos, Springer-Verlag, New York Inc., 1991.

[5] M. F. Barnsley and A. D. Sloan, “A better way to compress images,” BYTE magazine, pp215-233, Jan. 1988.

[6] M. F. Barnsley and S. Demko, “Iterated function systems and the global construction of Fractals,” Proc. Roy. Soc. London, Vol. A399, pp243-275, 1985.

[7] A. E. Jacquin, “Image coding based on a fractal theory of iterated contractive image transformations,” IEEE transaction on image processing, Vol. 1, No. 1, pp18-30, Jan. 1992.

[8] E. W. Jacobs, Y. Fisher and R. D. Boss, “Image compression: A study of the iterated transform method,” IEEE Transaction on signal processing, Vol. 29, No. 3, pp251-263, Dec. 1992.

[9] M. F. Barnsley and S. Demko, “Iterated function systems and the global construction of Fractals,” Proceeding Roy. Soc. London, Vol. A399, pp243-275, 1985.

[10] T. K. Truong, J. H. Jeng, I. S. Reed, P. C. Lee, and A. Q. Li, “A fast encoding algorithm for fractal image compression using the DCT inner product,” IEEE Transaction on image processing, Vol. 9, No. 4, pp. 529-535, Apr. 2000.

[11] C. M. Lai, K. M. Lam, and W. C. Siu, “A fast fractal image coding base on kick-out and zero contrast conditions,” IEEE Transaction image processing, Vol. 12, No. 11, pp. 1398-1403, 2003.

[12] C. C. Wang, L. C. Lin, and S. H. Tsai, “Fast fractal encoding algorithm using the law of cosines,” IEEE Inter. mid-west symposium circuits and systems, pp. I233-I236, 2004.

[13] M. Ghazel, G. H. Freeman, and E. R. Vrscay, “Fractal-wavelet image denoising,” in Proceeding IEEE International Conference image processing, 2002, pp.836–839.

[14] M. Ghazel, G. H. Freeman, and E. R. Vrscay, “Fractal image denoising,” IEEE Trans. Image Processing., Vol. 12, no. 12, pp. 1560–1578, Dec. 2003.