DICTA2002: Digital Image Computing Techniques and Applications, 21--22 January 2002, Melbourne, Australia.

A Fast Fractal Image Coding Approach Employing Fuzzy Aggregation of Domain Blocks

Khaled M. Mahar and Mohamed. T. El-Sonni
Computer Engineering Department, P.O.Box 1029, AASTMT, Alexandria, Egypt,
Email: khmahar@aast.edu

Abstract

Fractal image compression is a class of image compression techniques that offer the advantages of high compression ratio and fast decoding. But the major obstacle of these techniques is the relatively long time needed for encoding the image due to the requirement of search for the best fitting domain block for each range block. Various methods have been suggested to overcome this problem without much loss of image quality. In this paper, such methods are reviewed and a coding scheme is suggested in which multiple domain blocks are used to approximate each range block. These domain blocks are around the range block and hence a reduction in search-overhead is achieved in the encoding process. A comparative study between our approach and other fractal coding techniques is also presented.

1. Introduction

Image data compression has become an important issue for the purpose of storage and transmission of images or sequence of images due to their large memory requirements. Most of the methods in use can be classified under the head of *lossy* compression. This implies that the reconstructed image is always an approximation of the original image. This to be effective, one should take the human visual system into account when designing the compression algorithm.

The most popular among these methods is the discrete cosine transform (DCT) which is the JPEG standard for still image compression [1]. However in the recent past some new methods have been investigated which perform as well or better than the DCT in most cases. Notable among them are the wavelet transform [2], and fractal image compression [3].

Fractal, or iterated function system (IFS), image compression was first suggested by Barnsley [4] in 1988. However the first automated compression scheme for real world images using IFS was developed by Jacquin [3] in 1990. During the last decade immerse research activity has taken place in this field. As a result, currently IFS image compression is considered to be comparable to the existing methods at high and moderate bit rates (0.5 to 1 bpp) and superior to most methods at low bit rate (<0.25 bpp).

Although decoding is simple and fast, the main disadvantage of the IFS scheme is that the encoding process is computationally very intensive. In the encoding step the image is partitioned into disjoint blocks (range blocks). For each range block, another block (domain block) is selected from the same image. The goal is to approximate the pixel intensities of the range block with those of a domain block. Because good approximations are obtained when many domain blocks are allowed, searching the pool of domain blocks is time-consuming. This problem has attracted a degree of attention and many methods are investigated to speed up the encoding process while retain a good image fidelity.

In this paper, a new encoding scheme is proposed to accelerate the image encoding process. It tries only searching the nearest domain blocks which contain the range block. However if the required fitted domain block can not be found, the algorithm generates a new block from the searched domain blocks. The contribution of each domain block to the new block is confined by the approximation error between it and the encoded range block.

The proposed process can be regarded as a fuzzy aggregation where each domain block contributes to the fractal code by a degree derived from a membership function. This membership function is generated to represent the approximation error between a range block and a candidate domain block.

This paper is organized as follows: section 2 introduces the mathematical principle of fractal coding. It also reviews some of the algorithms concerned by the reduction of the encoding complexity. Section 3 outlines the proposed coding scheme used in this paper and it also presents the coding results. In section 4, a comparative study between the suggested approach and other fractal coding techniques is presented. Finally section 5 concludes the paper.

2. The mathematical principle of fractal coding

There are four main mathematical concepts [5] underlying IFS image compressions: metric spaces, contractive maps, the contractive mapping fixed point theorem, and the collage theorem. In a metric space (χ,d), χ is the set of M×N matrices whose elements correspond to pixel values of images, and d is a distance measure or metric. The most used metric is the squared error. For example if $\mathbf{A}=<a_{ij}>$, $\mathbf{B}=<b_{ij}> \in \chi$ then

$$d(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{M}\sum_{j=1}^{N}(a_{ij}-b_{ij})^2. \quad (1)$$

A map ω: $\chi\rightarrow\chi$ is said to be contractive with contractivity s if it satisfies

$d(\omega(\mathbf{A}),\omega(\mathbf{B})) \leq s\, d(\mathbf{A},\mathbf{B})\ \forall\ \mathbf{A},\mathbf{B} \in \chi$, where $0\leq s<1$. (2)

The contractive mapping fixed point theorem ensures that ω has a unique fixed point -called the attractor of ω- and this fixed point can be found by iterations of ω. That is, there exist a unique fixed point $\mathbf{F}\in\chi$ such that for any initial point $\mathbf{P}\in\chi$

$$\mathbf{F} = \omega(\mathbf{F}) = \lim_{n\to\infty}\omega_0^n(\mathbf{P}) \quad (3)$$

The collection $\omega_1,\omega_2\ldots\omega_n$ of contractive maps on a metric space (χ,d) is called iterated function system (IFS). The role of IFS in image encoding can be explained as follows: Consider an image matrix $\mathbf{A}\in\chi$ and let

$$\hat{\mathbf{A}} = \omega(\hat{\mathbf{A}}) = \lim_{n\to\infty}\omega_0^n(\mathbf{P}) \quad \forall\mathbf{P}\in\chi, \quad (4)$$

where ω is a contractive transformation, and $d(\mathbf{A},\hat{\mathbf{A}})$ is small. Thus if ω can be determined and represented more compactly than **A**, then it is called the compressed data for **A.** Therefore **A** is compressed. Unfortunately, it is difficult to construct ω directly but the collage theorem simplifies this problem to a large extent. It states that if $d(\mathbf{A},\omega(\mathbf{A})) < \varepsilon$ then $d(\mathbf{A},\hat{\mathbf{A}})$ is also bounded, where $\hat{\mathbf{A}}$ is the fixed point of ω. Usually ω is of the form

$$\omega = \bigcup_{i=1}^{N}\omega_i \quad (5)$$

where N is the number of range blocks and ω_i is a contractive mapping for the range block i. We use the collage theorem to optimize the parameters of the mapping for each ω_i separately. Without the collage theorem we would have to do a simultaneous optimization for all the mappings, which would be impractical.

2.1 Fractal image compression

The basic idea of fractal image compression is to find self-similarity between larger parts of an image and its smaller parts. This is accomplished by partitioning the original image into non-overlapping small portions (*range blocks*). Then trying to find various, at least very similar, matching from the larger parts (*domain blocks*) of an image.

Assume that the image is scaled to the unit square I^2 = [0,1] × [0,1] and the dynamic range of the gray level is contained in the interval I=[0,1]. The parametric form of the transformation ω_i: $I^3\rightarrow I^3$, i=1…Nr is chosen as

$$\omega_i\begin{pmatrix}x\\y\\z\end{pmatrix} + b_i = \begin{pmatrix}a_i & b_i & 0\\ c_i & d_i & 0\\ 0 & 0 & s_i\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix} + \begin{pmatrix}e_i\\f_i\\o_i\end{pmatrix}, \quad (6)$$

where (x,y) denotes the coordinates of a point in I^2 and z= g(x,y) denotes the intensity or grey level at (x,y). Associated with each ω_i is a range block $\mathbf{R}_i\in I^2$ and a domain block $\mathbf{D}_i \in I^2$. The N_r range blocks are non-overlapping and tile the unit square I^2 completely. The transformation ω_i and the domain blocks $\mathbf{D}_i$ are chosen to satisfy the following conditions:

- ωi maps Di into Ri,
- the size of each Di is larger than that of the corresponding Ri to ensure contractivity, and
- the collage error, or the squared error, between the Di and the corresponding Ri is minimum.

The parameters a_i, b_i, c_i, and d_i, determine the spatial contraction factor and the isometric operation performed on the block. The parameters e_i and f_i give the address of the domain block with reference to the range block. The parameters s_i and o_i are the intensity scaling factor and the offset respectively. This collection of parameters is called the IFS code for the image and if it is applied iteratively on any arbitrary initial image, the image converges to the fixed point, which is an approximation of the encoded image. The closeness of the reconstructed image to the original image is usually represented by the peak signal to noise ratio (PSNR). The PSNR of $\hat{\mathbf{A}}$, the approximation of **A**, is given by

$$\text{PSNR} = 10\log_{10}\frac{(2^b-1)^2 MN}{d(\mathbf{A},\hat{\mathbf{A}})} \quad (7)$$

where b is the number of bits used to code the pixel values of **A**, M and N define the image size**,** and the metric d is the squared error defined in (1).

3. Reduction of the encoding complexity

Due to the unacceptably high complexity of encoding of the FIC scheme, a lot of work has been directed at reducing the encoding complexity [6]. The methods suggested may be classified as

- Domain pool restrictions,
- Categorized search,
- Adaptive clustering,
- 1-D function methods,

- Feature vectors, and
- Transform Domain Block Matching.

3.1 Domain pool restriction

The first step in the reduction of computational complexity of the encoding process is to restrict the domain pool. Not all possible domain blocks are searched. Jacobs et [7] suggested restriction of the domain pool to blocks twice the size of the range blocks, with adjacent blocks overlapping one another by the size of the range blocks. Doing so brings the search time within reasonable bounds. But because not every domain block is under consideration, the optimal pairing for a given range block may be overlooked. Consequently, image quality suffers.

Another approach is to restrict the search to nearby area [8]. For example, the source image may be sectioned into four quadrants. For a range block in one quadrant, only domain blocks in that same quadrant are searched. As a result, the time complexity is reduced but this restricted area search depends on an image possessing locality of similar form but this cannot be guaranteed.

Beaumont suggested an outward spiral originating from the current-range-block position [9]. But instead of examining all candidate domain blocks for the best match, search halts as soon as a sufficiently good match is found. In many cases the search time is dramatically reduced, albeit with some loss of image fidelity.

3.2 Categorized search

Taking a different approach Jacquin [3] used a classification scheme to classify the domain and range blocks as shade, edge and mixed blocks. Edge blocks are further classified as simple and mixed edge blocks. For a range block from any one of the above categories, only domain blocks from the same category are searched.

A more elaborate classification scheme was proposed by Fisher and Boss [7]. In their approach, each domain block is inserted into one of 72 categories. To do so, a block is first divided into four quadrants and oriented (by flipping or rotation) so that the block falls into one of three major classes. If A_i, i=1 4 are mean pixel intensities of the four quadrants of a block, then the three major classes are:

Class 1: $A_1 \geq A_2 \geq A_3 \geq A_4$

Class 2: $A_1 \geq A_2 \geq A_4 \geq A_3$

Class 3: $A_1 \geq A_4 \geq A_2 \geq A_3$

The ordering of brightness of the quadrants for the three major classes is illustrated in figure 1. Once divided into these three major classes, the quadrants of each square are ordered from highest variance to lowest, for 4! =24 possibilities within each class. A range block is also categorized in this manner. When seeking a matching domain block, only the corresponding category is searched.

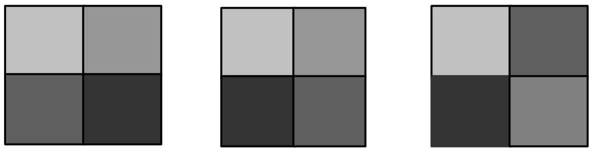


Figure 1. The three major classes of Fisher s classification scheme

Boss and Jacobs [10] developed an archetype classification scheme. In this scheme, the archetype form for a particular codebook block is given by the block that can best cover all others having the same archetype. Starting from an arbitrary classification using codebook blocks from a library made from many images, the classification scheme is iterated till the best archetype form for each block is determined. The iteration is stopped when there is no more change in the selection of archetype blocks in further iterations. These archetype blocks, derived from training images are used for classification of the range and domain blocks of image.

3.3 Adaptive clustering

In the three methods described earlier, the classification scheme is decided upon before it is applied to any image. On the other hand in the adaptive clustering scheme proposed by Oien and Lepsoy [11], the classification is image dependent. In this method, the range and codebook blocks are divided into disjoint sets

iteratively updated if any block does not fit into any particular cluster, at the same time satisfying some conditions.

3.4 1-D functional methods

Bedford *et al* [12] proposed a method, in which the range and domain blocks are compared by projecting each of them on a common, fixed unit vector. A particular range block is compared only with domain blocks whose result of comparison with the unit vector is close to that of the result of comparison of the range block. This method has further extended in [6] to include more such vectors.

3.5 Feature vectors

This method was first proposed by Saupe [13]. In this method, small sets of d real-valued keys are assigned to each range and domain block, which make up the d-dimensional *feature vector.* These keys are constructed such that searching in the domain pool is restricted to a small neighborhood around the d-dimensional key corresponding to a particular range block. Thus the sequential search of the domain block is substituted by a nearest neighbor search.

Kominek [14] used a much simpler scheme for arriving at the feature vector, but used the r-tree algorithm for searching. In this algorithm the d-dimensional space of the feature vector is divided into d-dimensional rectangles. For a range block with its feature vector located in one rectangle, only domain blocks with feature vectors in the same rectangle are searched.

3.6 Transform domain block matching

Wohlberg *et al* [15] proposed a scheme where block matching is done in the DCT domain. Here they make use of the well-known energy compaction property of the DCT to reduce the number of dimensions of the feature vector space, where a nearest neighbor search is performed. Also the fact that the DCT of the transformed blocks can be obtained from the DCT of the original block with just multiplication by ± 1 is put in use. Saup *et al* proposed a similar method where a fast search is performed via fast convolution [16]. The goal was to reduce the computational costs of the calculation of the inner products between the range blocks and codebook blocks (blocks formed from the domain blocks). This calculation can be carried out more efficiently in the frequency domain when the range block is not too small.

4. The proposed method

The proposed method starts with the partitioning of the entire image into a set of mutually disjoint range blocks. For each range block we consider a pool of domain blocks twice the linear size. These domain blocks are shrunken by pixel averaging to match the range block size. To speed up the encoding process the proposed method reduces the searched space by restricting the pool of domain blocks to the 4 domain blocks encompassing the range block. These domain blocks contain the range block as one of their quadrant as shown in figure 2. Hence this method generate a self-affine system.

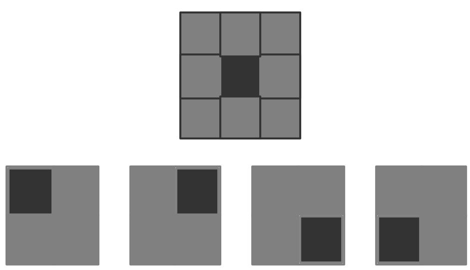


Figure 2. The relation between a range block (black boxes) and its four encompassing domain blocks.

The motivation for selecting this scheme is that a suitable fitted domain block may not be found even if we search anywhere in the image. To overcome this problem the algorithm tries to generate an appropriate block from the best-fitted domain blocks. The contribution of each domain block to the new block is confined by the approximation error between it and the encoded range block. This process can be regarded as a fuzzy aggregation where each domain block contributes to the fractal code by a degree derived from a membership function given by:

$$\mu = (1 - \mathrm{err})^2 \tag{8}$$

where err is a normalized error between a range block **R** and a domain block **D**.

The pool of domain blocks is enlarged before searching by including all 8 isometric versions (rotations and flips) of a block. This gives a pool of domain blocks $\mathbf{D}_1$ $\mathbf{D}_{ND}$. Assume that the blocks have been adjusted to zero-mean intensity level, and the decoding process starts from a low-resolution version of the original image derived from the range block averages. Then, the transformation used to approximate a range block **R** by a multiple domain blocks is given by:

$$\hat{\mathbf{R}} = \sum_{i=1}^{n} \mu_i (s_i \mathbf{D}_i) \tag{9}$$

where n is the number of best-fitted domain blocks used in the approximation, s_i is the intensity scaling factor, and μ_i is as specified in (8).

Before computing the compression ratio we should emphasize that the bit allocations in the fractal code for the position of the domain blocks are excluded. This is because the positions of the domain blocks can be concluded fom the position of the range block as shown in figure 2. Also, there is no need to store the scale factor that transforms a domain block to a range block since the domain blocks are twice the linear size of the range blocks. Let the range block averages represent the offset parameters o_i, and m_i denotes the 8 isometric operations applied, then the compression ratio can be derived from (9) using the following bit allocations:

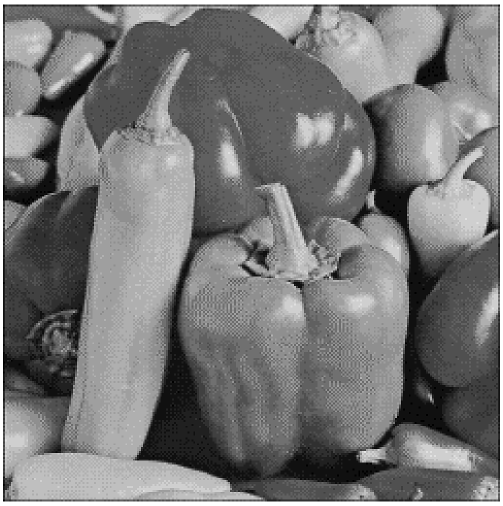
m_i 3 bits- 8 isometric operations
s_i 5 bits- sufficient from empirical tests [17]
o_i 6 bits- sufficient from empirical tests.

Only the 8 bits of m_i and s_i are replicated for each domain block used in (9). For example, if we use 8×8 range blocks and four domain blocks to approximate each range block then we need 32 bits for m_i and s_i plus 6 bits for o_i. By this way we achieve a respectable degree of compression ratio of 16:1.15.

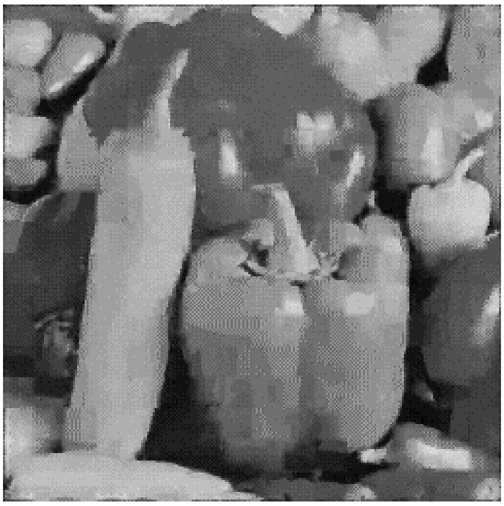
5. Experimental results

A number of standard images have been used for the test of the proposed method. One of these images is the

256×256×8 standard image "Pepper" which is shown in figure 3(a). We have compressed the image by using one-level partitioning. The size of the range block is 8×8, and the domain block size is 16×16. The decoded image with its compression ratio and the PSNR is shown in figure 3(b). The experiment shows two facts. First the encoding time is greatly reduced by the proposed method due to the local search around the range blocks. However, this local search limits the fidelity of the reconstructed images as inferred from the PSNR. Second although the compression ratio is comparable to the recent reported one, the image fidelity suffers from this high compression and the quality of the image can be enhanced to some magnitude if we consider smaller range block sizes.



(a) Original image



(b) Reconstructed image.

Figure 3. (a) The original 256×256×8 standard Pepper image, (b) Reconstructed image with compression ratio of 16:1.15 and PSNR equal to 22.6

6. Conclusions

We have presented a framework for fractal image encoding based on multi-domain blocks for approximating each range block. The contribution of each domain block to a range block is confined by the error between the original range block and this domain block. The process can be regarded as a fuzzy aggregation where each domain block contributes to the fractal code by a degree derived from a membership function. After trying and analyzing the results of our method, we have concluded that the limited domain pool affects the performance of the encoder in two ways. Due to the limited number of searched domain blocks, the method succeeds in reducing the encoding time. Also, because all the domain blocks are around the range block, there is no need to store their locations and hence our method achieves a high compression ratio. On the other hand, the restricted domain pool makes the quality of reconstructed images moderate. Future works will be toward the enhancement of the image fidelity through an optimum partitioning while keeping the local search strategy.

References

[1] G.K Wallace, 'The JPEG Still Picture Compression Standard', *Commun. ACM*, 34, pp 31-44, Apr 91.

[2] I.M Shapiro, 'Embedded Image Coding Using Zero-Trees of Wavelet Coefficients,' IEEE Trans. On Signal Proc. 42(12), pp. 3445-62,Dec. 93.

[3] A.E Jacquin, 'Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations,' IEEE Trans. On Image Proc. 1(1), pp. 18-30,Jan 92.

[4] M. Barnsley, Fractal Everywhere, Academic Press, San Diego, CA, 1988.

[5] Y. Fisher, 'Mathematical Background', in Y.Fisher (ed), Fractal Image Compression: Theory and Applications, Chapter 2, Springer Verlag, NY. 1995.

[6] D. Saupe and R. Hamzaoui, ' Complexity Reduction Methods for Fractal Image Compression' IMA Conf. In Image Proc.: Mathematical Methods and Applications, T.M. Blacklegde (ed), Oxford University Press, 1995.

[7] E.W. Jacobs, Y. Fisher, R.D. Boss, 'Image Compression: A study of the Iterated Transform Method, Signal Processing, 19,pp. 251-63,1993.

[8] J. Kominek, 'Advances in Fractal Compression for Multimedia Applications' University of waterloo Technical Report 1995.

[9] J. Mark Beaumont, 'Advances in Block Based Fractal Coding of Still Pictures' Proceedings of IEEE

Colloquium: The Application of Fractal Techniques in Image Processing, pp. 3.1-3.6,1990.

[10] R.D. Boss and E.W. Jacobs'Archtype Classification in an iterated Transform Image compression algorithm' in Y.Fisher (ed), Fractal Image Compression:Theory and Applications, Chapter 4, Springer Verlag, NY. 1995.

[11] S. Lepsoy and G.E. Oien, 'Fast Attractor Image Coding by Adaptive Codebook Clustering' in Y.Fisher (ed), Fractal Image Compression:Theory and Applications, Chapter 9, Springer Verlag, NY. 1995.

[12] T. Bedford, F.M. Dekking and M.S. keane, 'Fractal Image Coding Techniques and Contractive Operators' Nieuw Arch. Wisk(4) 10,3, pp. 185-218,1992.

[13] D. Saupe, 'Breaking the time Complexity of Fractal Image Compression' Technical Report 53, Institut fur Informatik, Universitat Frieburg, 1994.

[14] J. Kominek, 'Algorithm for Fast Fractal Image Compression' Proc. Of SPIE Digital Video Compression: Algorithms and Technologies, 2419, pp. 296-305, 1995.

[15] B.E. Wohlberg, G.De Jager, 'Fast Image Domain Fractal Compression by DCT Domain Block Matching, Elect. Lett. 30, pp. 474-75, Mar 94.

[16] D. Saupe, A new view of fractal image compression as convolution transform coding, IEEE Signal Processing Letters 3,7 (1996).

[17] G. Oien, 'Parameter Quantization in Fractal Image Coding' ICASSP-94, pp. 142-146. 1994.