COMPUTATIONAL & APPLIED MATHEMATICS

Volume 24, N. 1, pp. 83–98, 2005
Copyright © 2005 SBMAC
ISSN 0101-8205
www.scielo.br/cam

Fractal coding based on image local fractal dimension

AURA CONCI and FELIPE R. AQUINO

IC – Computer Institute, Department of Computer Science
UFF – Federal Fluminense University, r. Passo da Patria 156, 24210-240 Niteroi, RJ, Brazil
E-mails: aconci@ic.uff.br / felipe@lps.ufrj.br

Abstract. Fractal codification of images is based on self-similar and self-affine sets. The codification process consists of construction of an operator which will represent the image to be encoded. If a complicated picture can be represented by an operator then it will be transmitted or stored very efficiently. Clearly, this has many applications on data compression. The great disadvantage of the automatic form of fractal compression is its encoding time. Most of the time spent in construction of such operator is due on finding the best match between parts of the image to be encoded. However, since the conception of automatic fractal image compression, researches on improvement of the compression time are widespread. This work aims to provide a new idea for decrease the encoding time: a classification of image parts based on their local fractal dimension. The idea is implemented on two steps. First, a preprocessing analysis of the image identify the complexity of each image block computing its dimension. Then, only parts within the same range of complexity are used for testing the better self-affine pairs, reducing the compression time. The performance of this proposition, is compared with others fractal image compression methods. The points considered are image fidelity, encoding time and amount of compression on the image file.

Mathematical subject classification: Primary: 68P30; Secondary: 94A08, 68U10, 68U05, 68P20, 65D15.

Key words: image coding system, fractal compression, image compression, PIFS codes, fractal dimension.

1 Introduction

Nowadays, the hardware to capture images turns possible the use of digital images with great tonal and spatial resolution. Image files now have a tendency to

#596/04. Received: 05/I/04. Accepted: 09/XI/04.

become too large if these possibilities must be used for improve the information quality (for example in biomedical or satellite applications). Moreover, the relatively low price of digital cameras and multimedia systems associated with the growth of the image databases and the intense use of nets enhance the use of image files.

A major objective of image coding is to represent digital images with as few bits as possible while preserving the level of intelligibility, usability or quality required for the application. Image compression has two great purposes: reduction of time for image transmission and reduction of storage area. The level of information quality required vary widely, depending on the special tasks. Several are the forms used to compact data, many of them can be used to compress images if after compression and reconstruction it must be exactly equal to the original [16]. Image coding techniques which do not destroy any information and which allow exact reconstruction of the original digital data are said to be information-preserving (or lost free). In such applications as image analysis from space programs and medical diagnosis, exactly regeneration of the original data is expected.

However, in applications such cellular phone image transmission, image retrieval in database or advertising on the WWW, some information in the original image may be destroyed without sacrifice the intelligibility to human viewer [14]. The more quality it is possible to sacrifice, the lower will be the required bit rate for transmission or storage. Image compression techniques which do not aim exact reconstruction of the original digital data are said to be lossy techniques [16]. These have been developed to exploit both the inherent characteristics of images and properties of human vision.

Image data are spatially correlated and preset local statistical self-dependence. Much redundancy and noise exist on picture material. So it is possible to reduce the image size considering the removal of these redundancy. It is more central to the human visual system to perceive and interpret, in a sense, the scene then really see the image information. For example only the outlines of an object (like a tree) are important for its identification. How humans can associate few draw elements with images is not clear and new researches have appeared in this direction. But this can be exploited in image compression improving the compression ratio.

Fractal compression employs the above mentioned aspect. It is based on self-similar and self-affine sets. The codification process consists of the construction of an operator which will represent the image to be encoded [1]. If a complicated image can be represented by an operator then the picture will be transmitted or stored very efficiently. The great disadvantage of the automatic form of fractal compression is its encoding time [2]. Most of the time spent in construction of such operator is due on finding the best match between parts of the image to be encoded (Figure 1). However, since the conception of automatic fractal image compression [11], researches on improvement of the compression time are widespread [7].

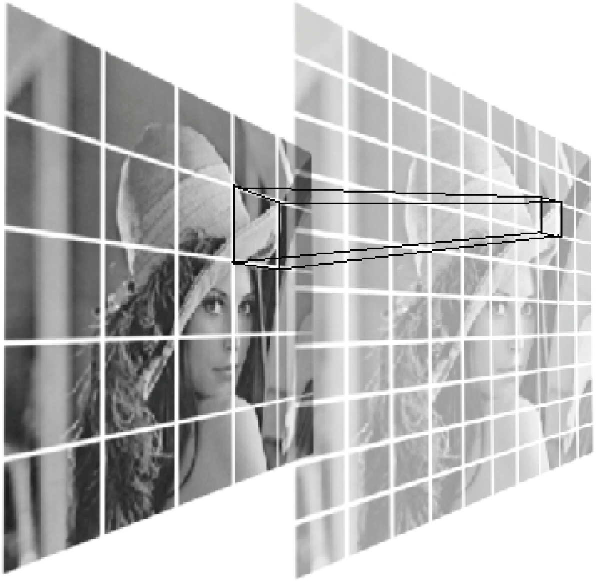

Figure 1 – In Partitioned Iterated Function System, self-similarity is sought between larger potions of the image (Domain blocks) and smaller potions (Range blocks clusters).

This work aims to provide a new idea for decrease the encoding time: a classification of image parts based on their local fractal dimension. The idea is implemented on two steps. First, a preprocessing analysis of the image identify the complexity of each image block computing its dimension. Then, only parts within the same range of complexity are used for testing the better self-affine pairs, reducing the compression time. The performance of this proposition, is compared with others fractal image compression methods. The points considered in this comparison are image fidelity, encoding time and amount of compression on the image file.

2 Encoding using FD for clustering

The fractal coding of an image $f(N \times N)$, can be seen as an optimization problem [1]. To modeling this problem let E be the space of gray scale images. Then, we choose a space of functions $f : X \mapsto G$, where the set X is taken as the set of spatial coordinates of the image and G represents the set of intensity values of this image. Also, let d be a metric, such that (E, d) is a complete metric space. Now the optimization problem is find a contractive mapping T on (E, d) whose fixed point $f = T(f)$ exist and is unique (by the contractive mapping fixed point theorem) [2]. A detailed description of basic aspect of this theory and implementations of automatic approaches can be found in [5].

The decoding process consists of iterating the mapping T from any initial image until the iterates converge to the atractor – an approximation of the original image [1]. For complex images T is not a single function but a system known as Iterated Function System – IFS [2]. An extension for fully automatic compression programs is the Partitioned Iterated Function System – PIFS [11]. PIFS basic idea is simplify the problem of compress the whole image by partitioning the image into blocks and then seek self-similarity between larger parts and smaller parts. The nature of the PIFS is illustrades in figure 1. The encoding process consists of the construction of the operator T, which is a system of affine transformation, each one defined by matching the better couple of sets, formed from the original image [2].

Several researchers have taken up the challenge to improve the basic automated algorithm for compression [7, 8, 15]. Some of the main subjects addressed are compression ratio, time reduction and image fidelity [3, 12]. We show, in the folowing that, the time of compression can be improved without lost of the image fidelity considering the contractive factor and the local image complexity. Depending on the block complexity, different contractive factors have been used to form two groups on the image partitioned for compression [11].

Fractal dimensions (FD) have been used to obtain shape information and distinguish between smooth (small FD) and sharp (large FD) regions [6, 13, 14]. In this work we propose it to characterize local complexity in images. In other words, the partitioned image is preprocessed and each local FD is computed to generate separate pools depending on local image complexity. To derive the contractive transformation only the relation between elements on pools of same FD are considered (improving compression time and quality of reconstruction).

Let Ri and Dj be two subsets on f (the first called range is formed from partitioning $n \times n$ non-overlapping regions on X and the second called domain is formed also by subset of X but witch may overlap). Let $t_i : Dj \mapsto Ri$ be a contraction defined by matching the better $t_i(Dj)$ for each Ri, which means that t_i is chosen such that the distance $d\big(Ri, t_i(Dj)\big)$ is as small as possible (figure 2).

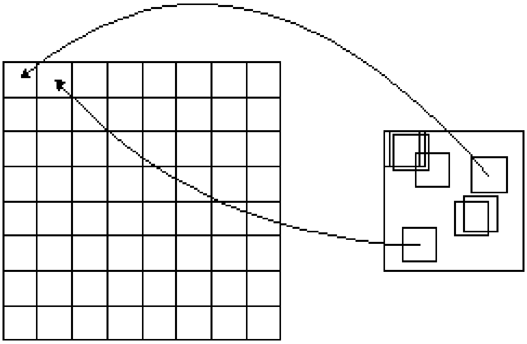

Figure 2 – Range blocks, reduced domain block and the formation of $t_i(Dj)$.

The operator T is given by

$$T_f = \bigcup_{i=1}^{\left(\frac{N}{n}\right)^2} t_i(f)$$

considering the contractive factor s of the Collage theorem [1] and fixed point A (attractor). Then

$$d(f, A) \leq \frac{d\Big(f, \bigcup_{i=1}^{\left(\frac{N}{n}\right)^2} t_i(f)\Big)}{1 - s}$$

that is the distance between the original image, f, and its reconstruction, A, the attractor of the PIFS, has (by the same theorem) an upper bound that is a function of the contractive factor s of T.

Several papers have popularized the scheme where a digital image is partitioned into square range blocks (say $n \times n$ pixels) and larger square domain blocks, frequently twice the size of the range blocks [8]. This scheme makes the compression process simple, but it uses the contractive factor of T equal to $s = \frac{1}{2}$. So by the Collage theorem [1, 2]:

$$d(f, A) \leq 2d\Big(f, \bigcup_{i=1}^{\left(\frac{N}{n}\right)^2} t_i(f)\Big)$$

We use $s = \frac{1}{4}$ or $s = \frac{1}{8}$ depending on the cluster – LFD. If the contraction of T is reduced to $s = \frac{1}{4}$, or $\frac{1}{8}$ then, by the theorem, the upper bound to $d(f, A)$ will decrease to:

$$d(f, A) \leq \tfrac{4}{3}d\Big(f, \bigcup_{i=1}^{\left(\frac{N}{n}\right)^2} t_i(f)\Big) \quad \text{or} \quad d(f, A) \leq \tfrac{8}{7}d\Big(f, \bigcup_{i=1}^{\left(\frac{N}{n}\right)^2} t_i(f)\Big)$$

and it is expected that the attractor, A, should look (each time more) quite like f, with these new set of contractions. This is the main idea of our work: the utilization of variable contractive factor, based on image complexity values.

Let D be a collection of subsets of from which D_j are chosen. D consistes of $(m \times m)$ pixels. The number of elements on D determines how much computation time the encoding takes on finding $t_i(D_j)$ for each R_i. For encoding, the mapping t_i must be specified and the better domain squares D_j must be chosen from a set of potential candidates. The choice of the domain pool as $4n \times 4n$ or $8n \times 8n$ for each $n \times n$ range block (figures 3 and 4), reduces the computation required for the search and minimizes image quality reduction [2].

We use local fractal dimension (represented by LFD) to subdivide the blocks into classes of complexity (see figure 5). Some methods on LFD estimation do not give satisfactory results in all possible range for images (from 2 to 3). For fast LFD evaluation a simple and efficient algorithm has been used [6]. We use two levels of complexity, based on the LFD of each image part: $2 \leq LFD < 2.5$ and $2.5 \leq LFD \leq 3$.

On finding each t_i, the domain sets D_j must be chosen from a set of potential candidates, *i.e.* a set with the same *LFD* of the range sets, R_i. Although, real images present semi-fractal behavior, they are considered in this analysis ideal fractal models. Of course it must be considered when one is addressing to the problem of finding the real fractal dimension of the image, but here it is only a way for representing its local complexity [13].

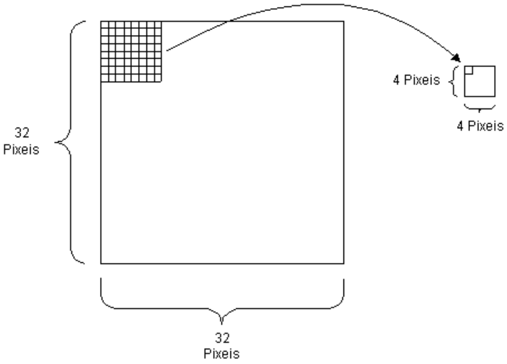

Figure 3 – Each pixel in the reduced domain block is forming averaging 4 × 4 pixel on the domain pool for*LFD* ∈ [2, 2.5).

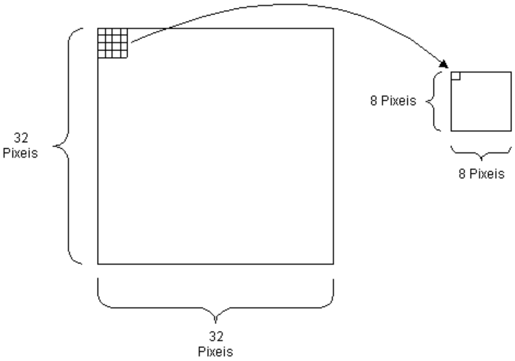

Figure 4 – Each pixel in the reduced domain block is forming averaging 8 × 8 pixel on the domain pool for*LFD* ∈ [2.5, 3].

3 Experimental results

Table 1 presents the results of this proposition on compression and reconstruction of 13 images. The efficiency of this image compression approach is quantified by considering both image fidelity and time reduction.

In this table, for no hardware dependece and installation interference on the results, the parameter related to the time spending on encoding is relative to the average results obtained on the same 13 images using the classical approach [2] (see table 2). Image fidelity is evaluated using 3 measures: the root mean square

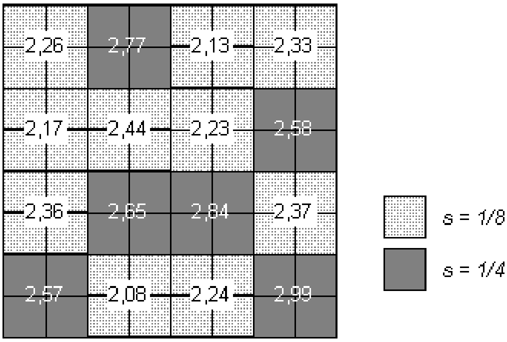

Figure 5 – Local fractal dimension blocks clusters.

Image name	$\frac{\text{Time}}{\text{Barnsley's Time Average}}$	e_{rms}	SNR_{rms}	$PSNR(dB)$
Lena	0.054413927	11.2595	9.0676	27.1005
Peppers	0.050105474	12.1646	7.4834	26.4289
Mandril	0.085676845	13.8837	8.3700	25.2807
LAX	0.077782276	18.0842	4.8113	22.9848
Cameraman	0.077108947	14.9070	7.9106	24.6630
Columbia	0.060836121	17.1769	5.1458	23.4319
Goldhill	0.053097100	9.8890	9.3313	28.2278
Couple	0.072218789	14.3472	7.9476	24.9955
Plane	0.066544502	16.5839	5.7887	23.7371
Woman	0.056708785	13.6920	8.1551	25.4015
Milk	0.046721783	11.3401	7.2516	27.0385
Man	0.064452068	14.4420	7.3699	24.9383
Lake	0.064435022	16.7244	2.6414	23.6638
Average	0.063853956	14.1919	7.0211	25.2225

Table 1 – Performance of the proposed method local fractal dimension with blocks of size $s = \frac{1}{4}$ and $s = \frac{1}{8}$ on the set of test images.

error, e_{rms}, the signal to noise ratio, *SNR* and the peak signal to noise ratio, *PSNR* of the each image. These are defined as:

$$
\begin{aligned}
e(x,y) &= \big(f(x,y) - A(x,y)\big) \\
e_{rms} &= \sqrt{\sum_{x=i}^{N}\sum_{y=i}^{N}\big(e(x,y)\big)^2} \\
SNR_{rms} &= \frac{\sum_{x=i}^{N}\sum_{y=i}^{N}\big(A(x,y)\big)^2}{\sum_{x=i}^{N}\sum_{y=i}^{N}\big(e(x,y)\big)^2} \\
PSNR &= 20\log_{10}\frac{2^p - 1}{e_{rms}}
\end{aligned}
$$

where $f(x,y)$ is the gray level on the original image, $A(x,y)$ is the gray level on the reconstructed image (attractor), N is the number of pixels in each image direcion and p is the number of bits per pixel used for definition of the gray level (128 pixels and eight bits deep in the tested images).

Figure 6 compares image quality on reconstruction of six images from the set of 13 analyzed: Lena, LAX, Cameraman, Columbia, Goldhill and Couple. All results obtained with these 13 pictures can be seen in http://www.ic.uff.br/~felipe/imgtest.html.

For comparison the same set is compressed by two other methods: the Heavy Brute Force Algorithm or Exhaustive Approach [2, 11] (table 2 and figure 7) and Restricted Area Search [12] (table 3 and figure 8). Figures 6-8 show also the result of subtraction: $e(x,y) = \big(f(x,y) - A(x,y)\big)$, where f is the original image and A its reconstruction (third and fourth columns). On third column, for better verification the absolute value of these subtractions are amplify by a scale factor of 5. On fourth column, an offset of 127 (the middle of the gray level) is used to turn possible visualization of negative or positive values.

Considering now the compression ratio. All images have 128 × 128 pixels (=16384 pixels) and 256 gray level. They need 131072 bits or 16384 bytes and 8 *bpp* (bits per pixel) before compression.

In the Heavy Brute Force algorithm or Exhaustive Aproach [2, 11], the domain blocks are always square and always twice the size of range blocks. The range blocks are 4 × 4 pixels and the domain blocks are 8 × 8 pixels. The number of

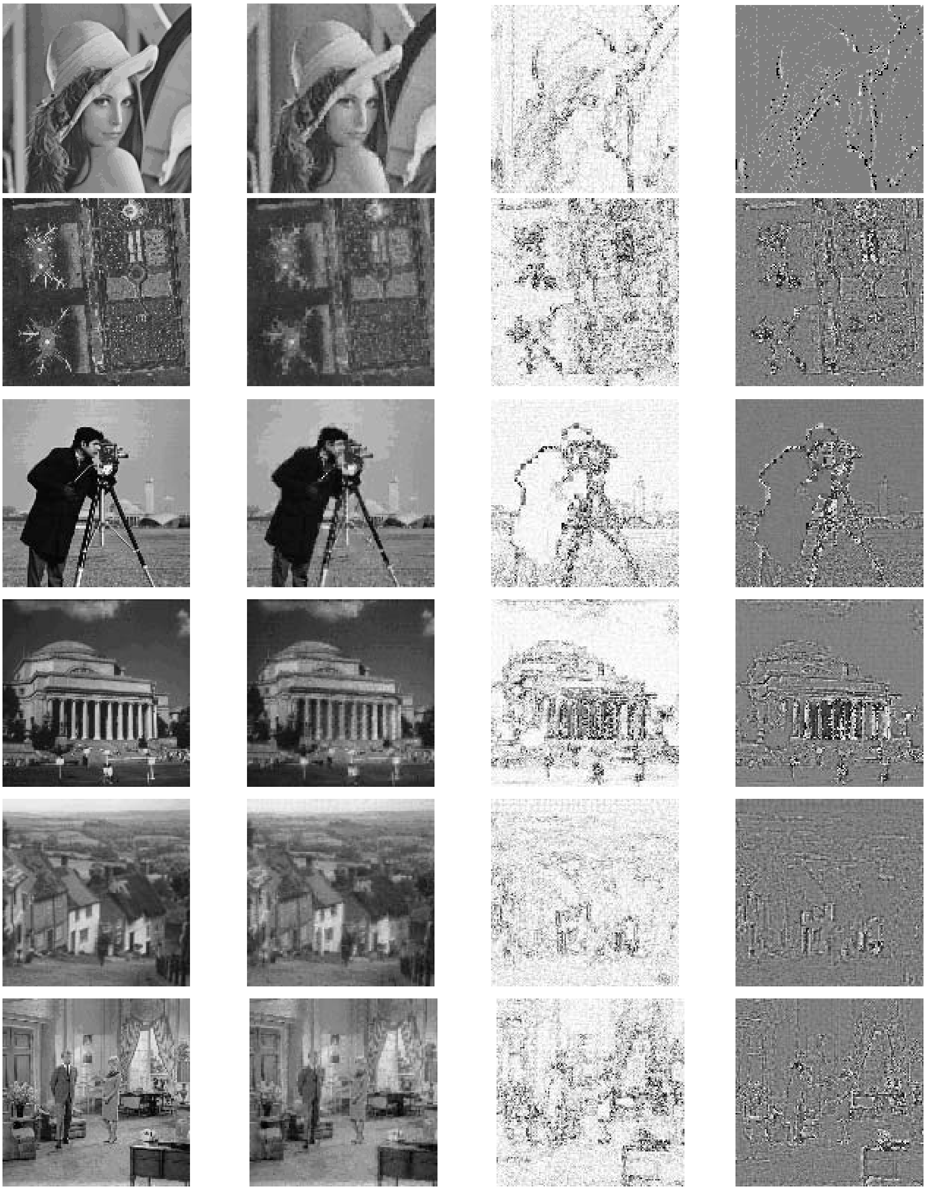

Figure 6 – Six images of the used experimented set of images (first column); reconstructed images using local fractal dimension with $s = \frac{1}{4}$ and $\frac{1}{8}$ (second column); difference image for quality verification: $5.\|e(x, y)\|$ (third column); and $\|5 \cdot \frac{e(x,y)}{2} + 127\|$ (fourth column).

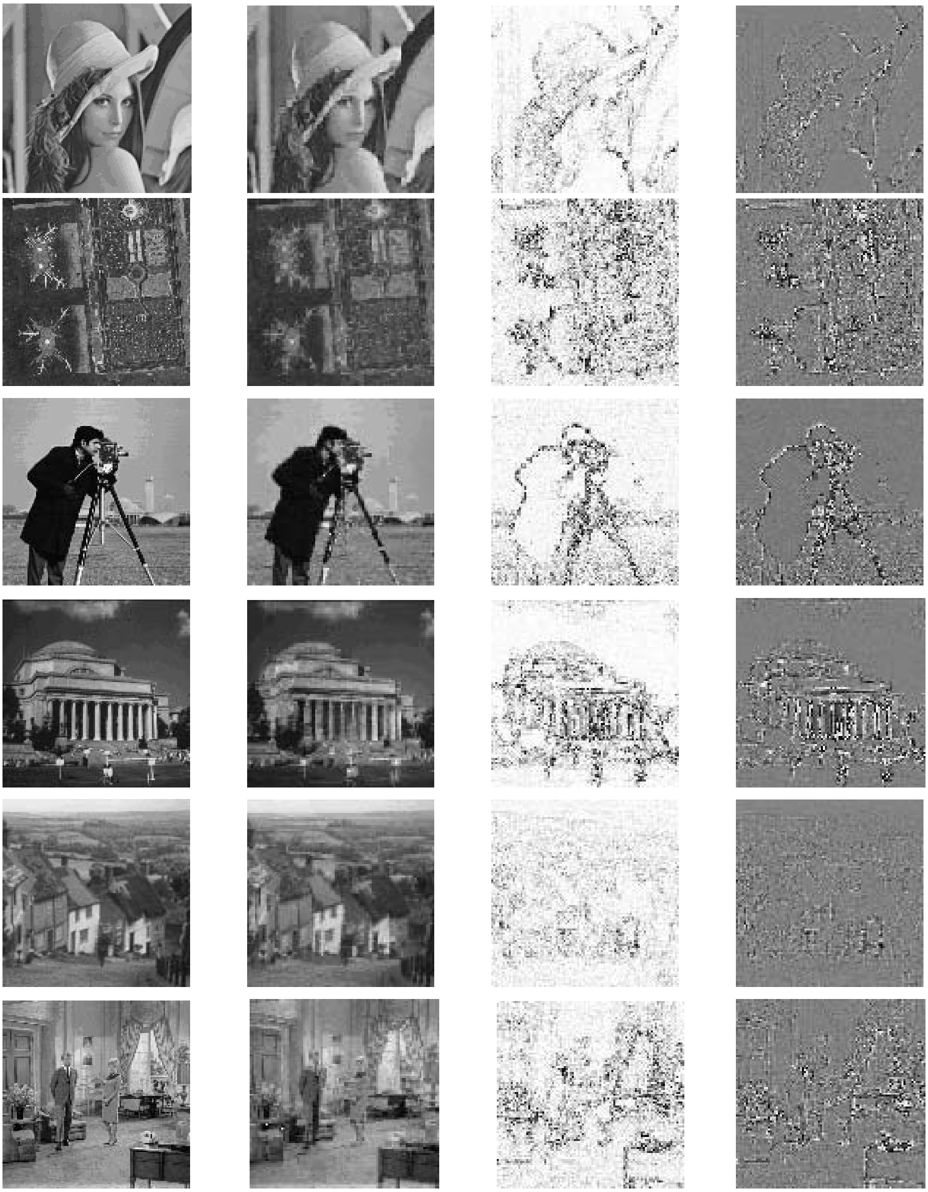

Figure 7 – Six of the images used in the experiments (first column); reconstructed images using Barnsley's program (second column); difference image for quality verification: $5 \cdot \|e(x, y)\|$ (third column); and $\|5 \cdot \frac{e(x,y)}{2} + 127\|$ (fourth column).

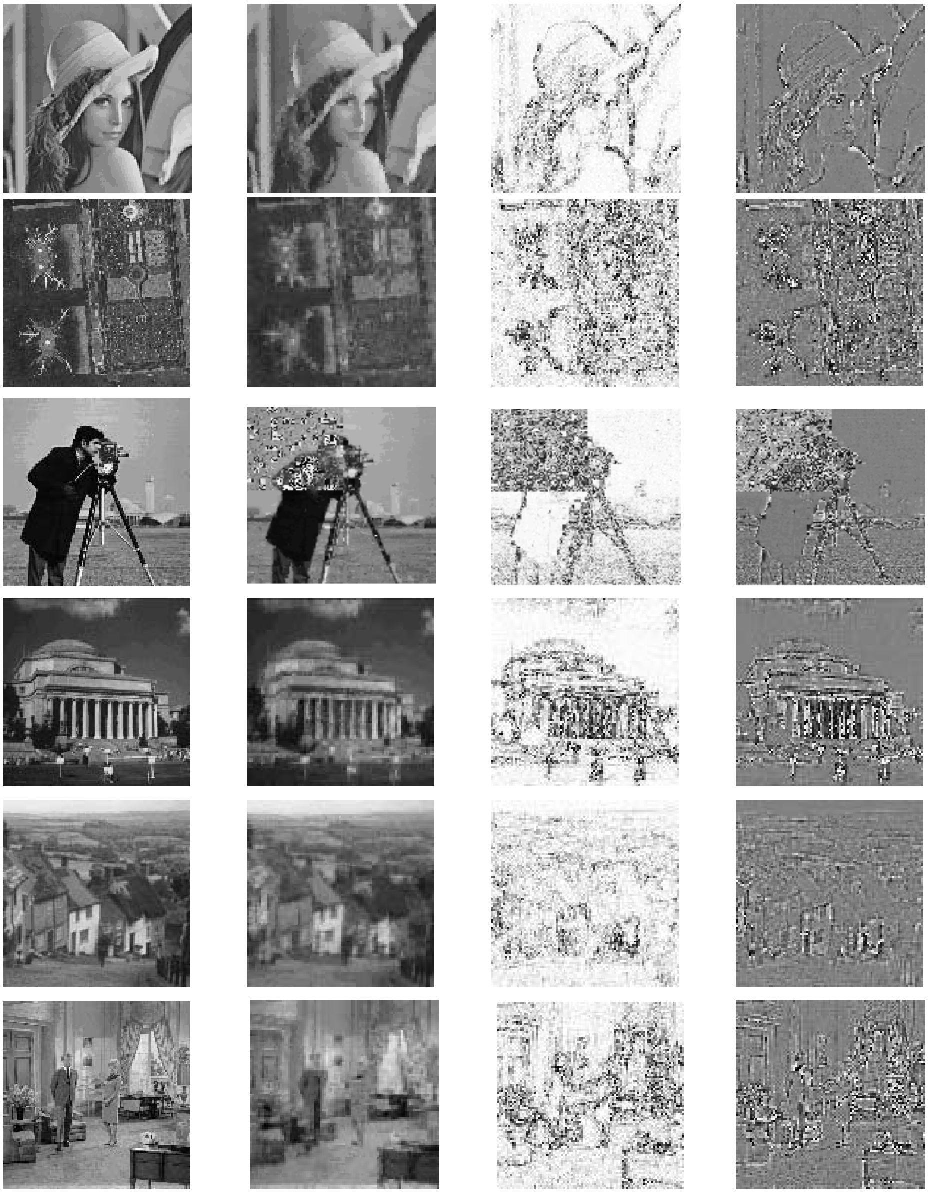

Figure 8 – Six test images used in the experiments (first column); reconstructed images using Restricted Area Search (second column); difference image for quality verification: $5 \cdot \|e(x, y)\|$ (third column); and $\|5 \cdot \frac{e(x,y)}{2} + 127\|$ (fourth column).

Image name	$\frac{\text{Time}}{\text{Time Average}}$	e_{rms}	SNR_{rms}	$PSNR(dB)$
Lena	1.000200294	7.61672	13.3478	30.4954
Peppers	1.000266348	7.5512	11.9571	30.5705
Mandril	1.000238648	13.1681	8.9190	25.7403
LAX	1.000215210	17.4734	4.9517	23.2832
Cameraman	1.000095885	14.0104	8.1885	25.2018
Columbia	1.000159809	16.3936	5.3475	23.8373
Goldhill	1.000138501	6.7977	113.4355	31.4836
Couple	1.000006392	13.6817	8.3402	25.4080
Plane	0.999230786	13.2835	7.5457	25.6646
Woman	1.000091624	11.0847	10.2303	27.2363
Milk	1.000093755	9.6735	8.4824	28.4191
Man	1.000025569	11.7124	9.3384	26.7579
Lake	0.999232916	15.8357	2.9538	24.1380
Average	1.000000000	12.1756	8.69527	26.7874

Table 2 – Performance of the Barnsley's program [2] for the same images.

block to be coded is $(128/4)^2$=1024. Each transformation can be coded using 23 bits (6 for locate de domain block in horizontal direction, plus 6 for locate the domain block in vertical direction, 3 bits for identification of the 8 symmetry and 8 bits to define the offset on intensity), then each compressed image can be stored as an text file of 23552 bits or 2944 bytes. On both, each image is represented by a file with 23552 bits. The compression ration for all images is $\frac{23552}{16384} = 1.4375\ bpp$ (bits per pixel).

In the proposed approach, one bit more is needed for represent what will be the cluster of the block. Then each block needs 24 bits for its storage as an text file, and each image can be represented as $1024 * 24 = 24576$ bits or 3072 bytes, in the more unfavorable case. So the worse compression ratio, using the proposed approach is $\frac{24576}{16384} = 1.5\ bpp$ (bits per pixel).

The Restricted Area Search [12] approach considers a reasonable way of reducing the encoding time to restrict the search to nearby areas. For example, the source image may be sectioned into four quadrants (figure 9). For range block in the bottom left quadrant, say, only domain blocks in that same quadrat are search. As a result, if the better pair is not in the same quadrant a poor quality of compression/reconstruction can be obtained (see and compare the Cameraman image before and after reconstruction using Resitricted Area Search on third line of figure 8).

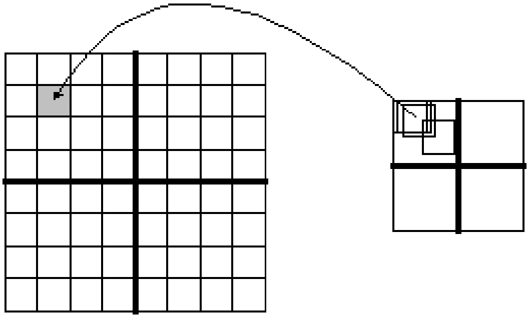

Figure 9 – Reducing the encoding time by restrict the search to nearby areas.

4 Conclusions

The principal goal in fractal compression has been maximum speed and quality. While still an open challenge, the presented approach is a significant advance in this direction. It is, on average, 15 time faster then early attempts (the Restricted Area Search is 4 times faster the classical approach). Moreover, when consider visual quality, the Error rms, the Signal to Noise Ratio rms and the Peak Signal to Noise Ratio there is not great difference between this and the classical approach. And it is clearly better than Restricted Area Search. The proposed aproach presents also no relevant increase on compression ratio.

In http://www.ic.uff.br/~aconci/compressao/fractal.html or http://www.ic.uff.br/~felipe/mdfl.html several methodologies suggested in other works (Light Brute Force, Local Sub Search, Local Search, Look Same Place Search, Double Canonic Classification, Canonic Classification through the Intensities and Double Canonic Classification with Quadtree of 3 levels) are compared with the here presented technique. Color images have also be tested (http://www.ic.uff.br/~felipe/imgcolor.html).

Image name	$\frac{\text{Time}}{\text{Barnsley's Time Average}}$	e_{rms}	SNR_{rms}	$PSNR(dB)$
Lena	0.227252775	8.6244	11.7174	29.4162
Peppers	0.227263429	9.0175	10.0652	29.0291
Mandril	0.227184590	14.2889	8.2306	25.0309
LAX	0.227284737	18.5487	4.6653	22.7645
Cameraman	0.227323091	41.6585	2.8115	15.7367
Columbia	0.227254906	17.7861	4.9788	23.1292
Goldhill	0.227254906	8.3060	11.0890	29.7430
Couple	0.227235729	14.3430	8.0140	24.9980
Plane	0.227242121	15.3609	6.3633	24.4025
Woman	0.227278345	12.8468	8.9592	25.9549
Milk	0.227167544	10.5219	7.7336	27.6889
Man	0.227233598	13.1114	8.3858	25.7778
Lake	0.227257037	17.3209	2.7885	23.3594
Average	0.227242121	13.3397	7.7492	25.9412

Table 3 – Performance of the Restricted Area Search's method [12] for the same images.

Acknowledgment

This work has been supported in part by the projects Protem-CC QualiIn and CNPq (Ref. 3024369/2004-9). The authors acknowledge the VisualLab (CTC-UFF) for the programming support.

REFERENCES

[1] M.F. Barnsley, *Fractals Everywhere*. 2nd ed., Academic Press, New York, (1993).

[2] M.F. Barnsley and L. Hurd, *Fractal Image Compression*. AK Peters, Wellesley, (1993).

[3] Chwen-Jye Sze, Hong-Yuan Mark Liao, Kuo-Chin Fan, Ming-Yang Chern and Chen-Kou Tsaoy, *Fractal image coding system based on an adaptive side-coupling quadtree structures*. Image and Vision Computing, **14** (1996), 401–415

[4] A. Conci and F.R. Aquino, *Using Adaptive Contraction for Fractal Image Coding Based in Local Fractal Dimension*. Proceedings of SIBGRAPI: the Brazilian Symposium on Computer Graphics, Image Processing and Vision, IEEE Computer Science, Campinas, SP, 231–239 (1999).

[5] A. Conci and F.R. Aquino, *Codificação Fractal de Imagens*. Proceedings of DINCON: I Congresso de Dinâmica, Controle e Aplicações, cap. 16, 365-402. Ed. by J.M. Balthazar, M. Boaventura, G.N. Silva e M. Tsuchida. SBMAC – ISBN-85-86883-05-0, São José do Rio Preto, SP (2002).

[6] A. Conci and C.F.J. Campos, *An Efficient Box-Counting Fractal Dimension Approach for Experimental Image Variation Characterization*. Proceedings of IWSIP'96 – 3rd International Workshop in Signal and Image Processing, Elsevier Science, Manchester, UK, 665–668 (1996).

[7] Y. Fisher, *Fractal Compression: Theory and Application to Digital Images*. Springer Verlag, (1994).

[8] J.C. Hart, *Fractal Image Compression and Recurrent Iterated Function Systems*. IEEE Computer Graphics and Applications, 25–40 (1996).

[9] R. Hamzaoui and D. Saupe, *Combining Fractal Image Compression and Vector Quantization*. IEEE Transactions on Image Processing. **9** (2000), 197–208.

[10] R. Hamzaoui, H. Hartenstein and D. Saupe, *Local Iterative Improvement of Fractal Image Codes*. Image and Vision Computing. **18** (2000), 565–568.

[11] A. Jacquin, *Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations*. IEEE Transactions on Image Processing. **1** (1992), 18–30.

[12] J. Kominek, *Advances in Fractal Compression for Multimedia Applications*. [ftp://links.uwaterloo.ca:/pub/Fractals/Papers/Waterloo/kominek95c.ps.gz] University of Waterloo, Internal report CS95-28 (1995).

[13] S. Morgan and A. Bouridane, *Application of Shape Recognition to Fractal Based Image Compression*. Proceedings of IWSIP'96-3rd International Workshop in Signal and Image Processing, Elsevier Science, Manchester, UK, 23–26 (1996).

[14] M. Nappi, G. Polese and G. Tortora, *FIRST: Fractal Indexing and Retrieval SysTem for Image Databases*. Image and Vision Computing. **16** (1998), 1019–1031.

[15] D. Saupe and R. Hamzaoui, *A Review of Fractal Image Compression Literature*. Computer Graphics, **28** (1994), 268–275.

[16] A. Watt and F. Policarpo, *The Computer Image*. ACM SigGraph series, Addison-Wesley, Essex, UK. (1998).