Using Adaptive Contraction for Fractal Image Coding Based on Local Fractal Dimension

AURA CONCI AND FELIPE R. AQUINO

IC- Instituto de Computação, CAA- Computação Aplicada e Automação, UFF - Universidade Federal Fluminense
R Passo da Pátria, 156, 24210-240 Niterói, RJ, Brasil
aconci@caa.uff.br, felipe@lps.ufrj.br

Abstract.. In automatic fractal image compression, most of the encoding time is spent on finding the best match between each range block and the domain blocks. We propose the use of the local complexity of the image domain blocks to reduce the number of pairs to be tested on this search. Indexing the contracted domain pools according to their local fractal dimension and using multiple contraction factors we can confine all potential matches to a relatively small number of possibilities. By selecting an appropriate criterion for close match, the compression time can be shortened without image quality degradation. The performance of the proposed algorithm, evaluated by means of fidelity versus encoding time and compression ratio, is compared with five approaches. Future developments can also improve the compression ratio by using fractal dimension in the definition of the size of range blocks.

Keywords: Fractal dimension, fractal compression, partitioned iterated function system-PIFS, image compression

1. Introduction

Image compression techniques are used for storing and transmitting images with as few bits as possible by removing redundant data [Weeks Jr. (1996)]. The number of applications using these techniques have increased due to requirements of rapid transmissions and reduced image storage space [Nappi et al., (1998)]. A lot of recent papers have addressed this topic [Rila (1998)], [Boulgouris--Strintzis (1998)], [Franssens et al. (1998)]. The compression algorithms can be divided into two categories: the lossless compression algorithms, and the lossy compression algorithms. The lossless compression algorithms guarantee the highest image quality at the expense of a low compression gain [Yun et al. (1997)]. The lossy compression algorithms give a high compression gain but the compressed image can differ from the original. The fractal technique belongs to the lossy compression algorithms.

Automatic fractal compression presents high compression ratio but the encoding time to generate the Partitioned Iterated Function System (PIFS) is high [Bansley (1993)]. This is due to the need of computing a large number of best matches among the two sets of image blocks, called range blocks and domain blocks [Fisher (1995)]. The most basic form of the encoder examines exhaustively every possible domain block for each range block [Jacquin (1992)]. As a consequence, the encoding time is a serious problem when the image is large due to the large number of blocks to be mapped [Hart (1996)]. Morgan and Bouridane (1996) addressed this problem by first classifying the image pieces (in four or sixteen classes) according to their high frequency components by using the Sobel operator, and thereafter compute the PIFS by searching the appropriate class. The local searching idea, used by Lu (1997) is related to the experiences that the best match is close when the position of the domain block is either chosen directly above, below, to the left, to the right of the range block. Lee and Lee (1998) present a method to reduce the search space based on the relationship between the local mean and variance of the range and domain blocks. Other methods require complex analyses on contours and textures of image blocks.

Our work is based on the fact that two blocks with very different complexities cannot be matched. Fractal dimension (FD) of surfaces could be used to obtain shape information and distinguish between smooth and rough regions. The image local FD (LFD) can be used for characterization of the block complexity. This implies that domain blocks whose FD differ greatly from the range blocks FD may be eliminated from the domain pool as candidates for matching. For each range block, only the domain blocks within the corresponding class of FD are considered. A major disadvantage is that the LFD of a photographic image is hard to estimate by usual computational methods. Moreover, many algorithms saturate before covering the range of possible values for such images, which is between 2 and 3 [Conci and Proença (1998)]. Sarkar and Chaudhuri (1994) described an efficient box-counting approach using the ε-blanket idea [Peleg et.al. (1994)], named Differential Box-Counting (DBC), which uses differences for computing the FD, and gives satisfactory results for the whole range. With some modifications, this algorithm can be used for efficient identification of the FD of images [Conci and

Campos (1995)] and with some improvements it becomes adequate for local block complexity computation [Aquino (1998)]. The complexity of each block is also related to the arduousness to well cover it using larger partitions. Moreover, the DF can be used to control the size of the image blocks, which is related to the number of pixels considered in each range block and the contraction factor of the maps. In fractal compression each range block is represented by an affine transformation of the encoded image that must be found by the encoder. Therefore LFD can improve the fractal encoding in many aspects.

In this paper we propose a new algorithm for compression using the above-mentioned idea. The organization of the paper is as follows. A concise review of fractal based coding followed by the presentation of the proposed algorithm is given in Section 2. Experimental results are detailed in Section 3 followed by the conclusions in Section 4.

2. Bases and Development

In general, image compression can be separated into two processes: the encoder or compressor and the decoder or decompressor. A grayscale image f ($N\times N$ pixels) can be seen as an element in the space E of functions

$$f : X \rightarrow G$$

where the set X is taken as the set of spatial coordinates of the image and G represents the set of intensity values of the image. A metric is used such that (E, d) is a complete metric space. The fractal coding of f can be seen as the problem to find the contraction operator $\boldsymbol{T}_f$ on (E, d) whose fixed point $f = \boldsymbol{T}_f(f)$ exist and is unique (Banach Fixed Point Theorem) [Kubrusly (1997)]. This is known as Iterated Function System (IFS) [Barnsley (1993)].

To find this operator $\boldsymbol{T}_f$ we define two sets on f : R_i and D_j (Figure 1). The set D_j , called domain, is formed from partitioning X into possibly overlapping regions. The set R_i , called range, is also formed from X but from partitioning X into non-overlapping regions with n×n pixels. For each R_i a contraction t_i: $D_j \rightarrow R_j$ is chosen such that the distance $\boldsymbol{d}(R_i,\ t_i(D_j))$ is as small as possible, considering all D_j. Each t_i is an affine transformation:

$$t_i(f) = t_i\begin{pmatrix} x_1 \\ x_2 \\ g \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ g \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ o \end{pmatrix}$$

Then the operator $\boldsymbol{T}_f$, or PIFS, is given by [LU (1997)]:

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

where $i = 1, 2, .., m=(N/n)^2$. A more detailed description of the encoding method, the notation used and descriptions of basic implementations can be found in (www.caa.uff.br/~aconci/compressao/fractal.html).

The fractal encoding process consists of the $\boldsymbol{T}_f$ construction, which is defined by matching the best couple of self-similar sets (R_i, D_j). This represents a compression of information when compared with the size of the set containing all pixel values from the image. Specially, if R can be composed by a large number of pixels from the original image (large range block). The disadvantage of the encoding process is that the number of pairs to be tested for each best couple is large, and each one has 8 possible space symmetries plus the change on contrast and brightness (elements o and s of t_i).

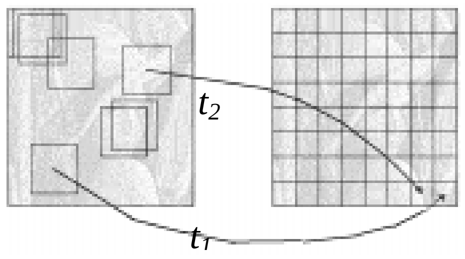


Fig. 1- Few domain image blocks, range image blocks and two transformations $t_1{:}D_j \rightarrow R_1$; $t_2{:}D_k \rightarrow R_2$.

Fractal decoding consists of iterating the mapping $\boldsymbol{T}_f$ from any initial image until the iterations converge to a fixed image, its atractor A (an approximation of the original image). Given the set of transformations, reproduction of the original image (decompression) is computationally simple and fast [Barnsley and Hurd (1993)].

The motivation for minimizing $d(R_i,\ t_i(D_j))$ is provided by the Generalized Collage Theorem [Fisher (1995)]: *Let $T_f = \cup_{i=1} t_i$ be a contraction on the complete (E, d) metric space with contraction factor s and fixed point A. Then*

$$d(f, A) \leq ((1 - s).(1 - \sigma))^{-1} (1 - \sigma^n)\, d(f,\ \cup_{i=1} t_i(f)),$$

where $i = 1, 2, ..., m=(N/n)^2$, σ is the Lipshitz factor of f and n is the exponent of eventual contractility.

The Collage Theorem gives an upper bound for the distance between, the original image, f and its attractor, A, as a function of the contraction factor s of T. Several works use an approach where a digital image is partitioned into square range blocks and square domain blocks of twice the size of the range blocks [Jacobs et al.(1992), Kominek (1995), Saupe and Hamzaoui (1994), Sze et al (1996)]. The consequence of using such approach is that any contraction factor of **T** is 1/2. So by the Collage Theorem:

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

If the contractility of the associated PIFS is reduced to s = 1/4, 1/5 or 1/6, then by this theorem the distance, d, between the original image, f, and the attractor, A, of the PIFS will decrease to

$$d(f, A) \leq 4/3\;\; d(f, \bigcup_{i=1}^{(N/n)^2} t_i(f))$$

$$d(f, A) \leq 5/4\;\; d(f, \bigcup_{i=1}^{(N/n)^2} t_i(f))$$

or to

$$d(f, A) \leq 6/5\;\; d(f, \bigcup_{i=1}^{(N/n)^2} t_i(f))$$

with these new set of contractions.

Our work utilizes the above described variable contraction factor, based on fractal complexity of image parts. For the complexity evaluation we compute the FD of 2×2 groups of range blocks, local fractal dimension-LFD, using a scheme developed in [Aquino (1998)] based on the algorithm described in [Conci and Campos (1995)]. We use three levels of complexity: 2≤LFD<2.33, 2.33≤LFD<2.66 and 2.66≤LFD≤3. We choose here to use only three ranges of LFD, others threshold values and numbers of division can be seen in [Aquino (1998)] and [Conci and Aquino (1999)].

The mapping t_i is specified by the better domain D_j chosen from a set of potential candidates called the domain pool. The number of elements in a domain pool determine the number of computations required for the encoder to find $t_i(D_j)$ for each R_i.. Choosing the size of the domain block as 4n×4n, 5n×5n or 6n×6n for each n×n range block, significantly reduces the computation of the search for the best self-similar matching, since the size of the domain pool to be searched for each range block is inversely proportional to the square of its scale factor. The number of pixels in each block is related to its local fractal dimension. The use of this approach will not have a significant effect on the image quality, as can be seen in the next section. Of course, other scales (1/2, 1/3, 1/8,...) can be used, which has been analyzed in [Aquino (1998)].

The compressor of the proposed algorithm has the following stages:

1. generate the domain pools considering the scale factors (Figure 2);
2. compute LFD of a group of 2×2 range blocks;
3. select the domain pool based on the LFD of the corresponding group of range blocks;
4. search the best reduced domain block for each range block inside the group of 2×2 range blocks, considering only the domain pool related with the LFD of the group;
5. store the information of the transformation for the best match;
6. return to stages 2 until the codification of all range blocks.

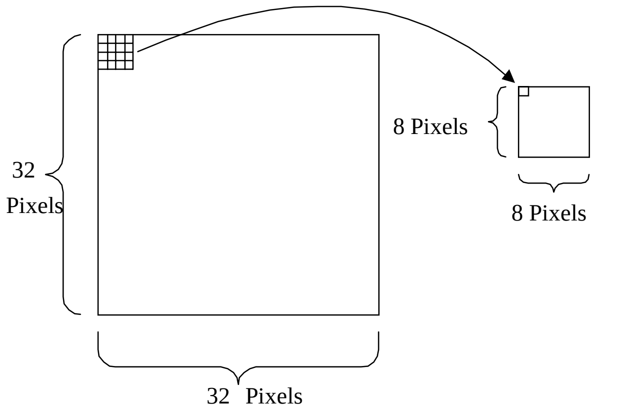


Fig. 2 - Each pixel on the reduced domain block is formed averaging the intensity of 4x4, 5x5 or 6x6 pixels on the domain block.

The number of pixels in the range blocks is important. In the original image each pixels is represented by 8 bits (256 gray level image) or 8 bpp (bits per pixel). In the compressed image each range block is represented by a fixed number of bits, which is related to the number of pixels used in the implementation to represent the affine transformation ti. Therefore the compression ratio is related to the number of range blocks. However, the number of pixels in the range blocks is dependent of the number of pixels of the image, because it will affect the quality of the decompressed image. Large images may have range blocks bigger than the ones of small images. Range blocks with large number of pixels tend to produce images with increased aliasing. Probably a good approach for improving the compression is to use LFD to define the number of pixels on the range block. This is beyond the scope of this paper and will be treated in future works.

3. Comparing Performances

Four images of different levels of complexities are used to evaluate the proposed algorithm (Table 1). Figure 3 shows original Milk, Lena, Goldhill and Peppers images. Figure 4 shows the reconstruction after compression by using our new algorithm and other approaches. To choose these images as samples we observed the main motive, its type (fine details, edges, lines), the number of elements, the richness of self-similarities, and the background.

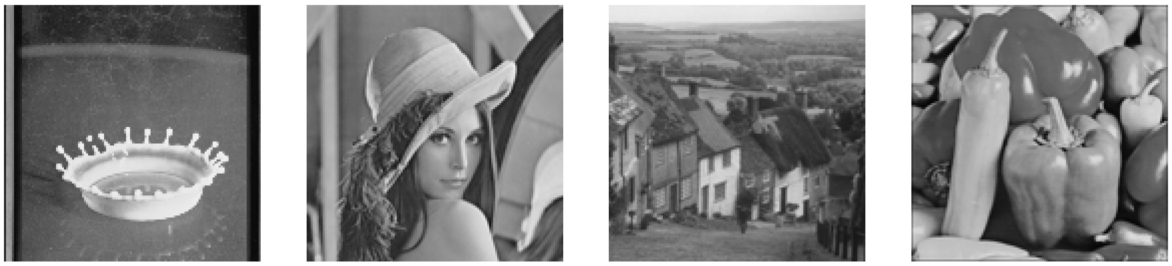
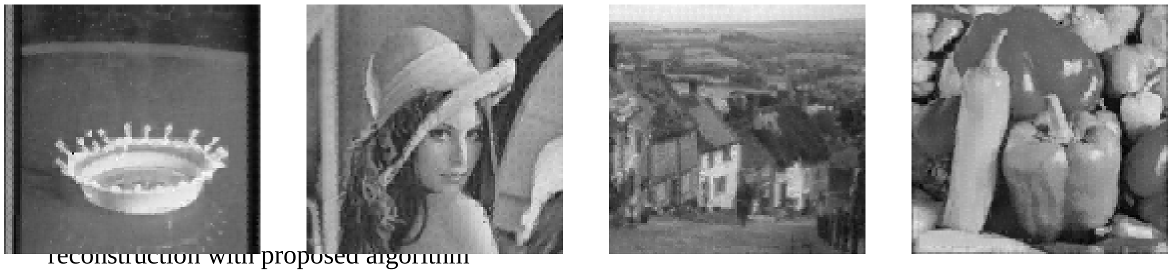


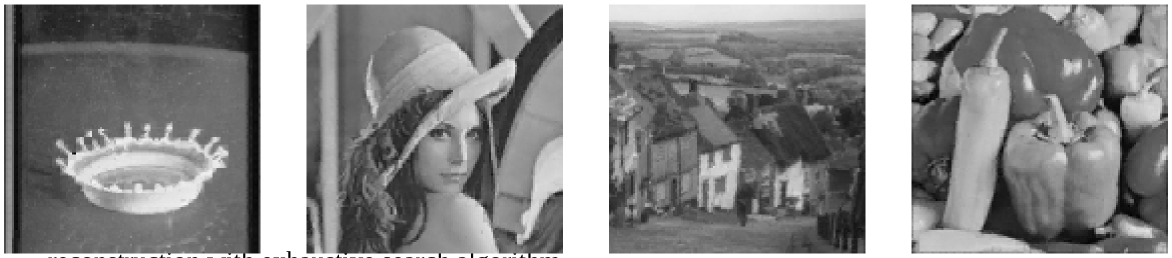
Fig. 3- Original images used on the proposed and other implemented algorithm

algorithm	image	Time (s)	e_{rms}	SNR rms	Compression ratio(bpp)
proposed	Milk	43.54	10.49	7.83	1.5625
	Lena	44.52	10.01	10.23	1.5625
	Goldhill	45.02	8.82	10.57	1.5625
	Peppers	44.74	10.22	8.78	1.5625
	average	44.45	9.88	9.35	1.5625
exhaustive search	Milk	469.35	9.67	8.48	3.5020
	Lena	469.40	7.61	13.34	3.5020
	Goldhill	469.37	6.79	13.43	3.5020
	Peppers	469.43	7.55	11.95	3.5020
	average	469.39	7.90	11.80	3.5020
look at same place	Milk	0.19	28.85	3.03	0.6875
	Lena	0.21	26.35	3.86	0.6875
	Goldhill	0.27	27.66	3.63	0.6875
	Peppers	0.21	29.02	3.26	0.6875
	average	0.22	27.97	3.45	0.6875
local search	Milk	9.58	13.78	5.90	1.4375
	Lena	9.14	10.50	9.55	1.4375
	Goldhill	9.60	10.35	8.87	1.4375
	Peppers	9.61	12.21	7.65	1.4375
	average	9.48	11.71	7.99	1.4375
restricted area search	Milk	106.61	10.52	7.73	1.4375
	Lena	106.65	8.62	11.71	1.4375
	Goldhill	106.65	8.30	11.08	1.4375
	Peppers	106.65	9.01	10.06	1.4375
	average	106.64	9.11	10.15	1.4375
scanning steps of 8 pixels range of 4x4 pixels	Milk	8.70	13.90	5.94	1.4375
	Lena	8.65	11.24	9.14	1.4375
	Goldhill	8.65	10.24	9.02	1.4375
	Peppers	8.23	11.95	7.61	1.4375
	average	8.56	11.83	7.92	1.4375

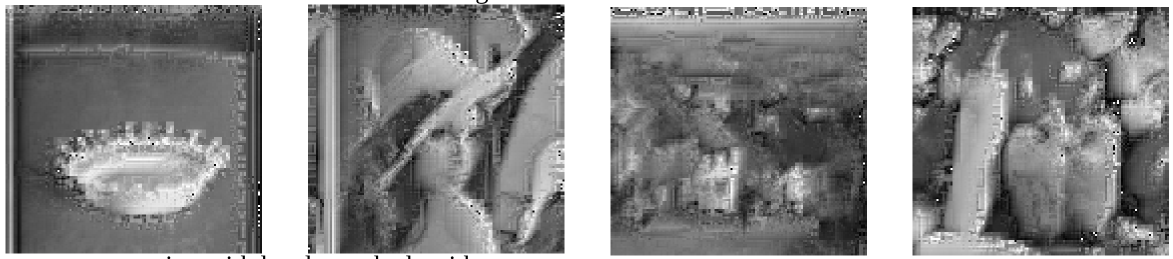
Table 1- Comparing features using the proposed and other implemented algorithms



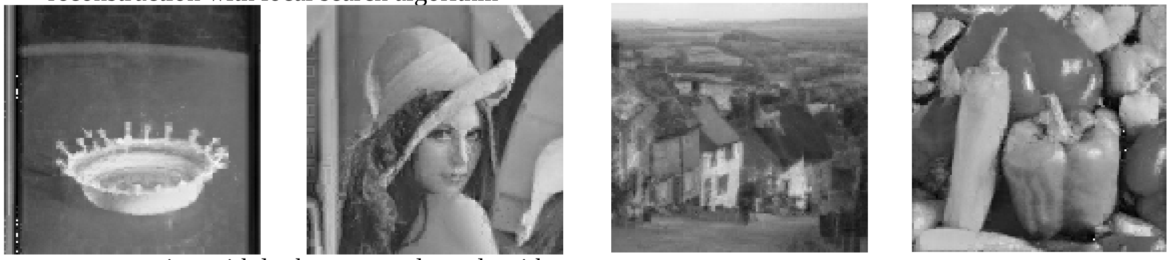
reconstruction with proposed algorithm



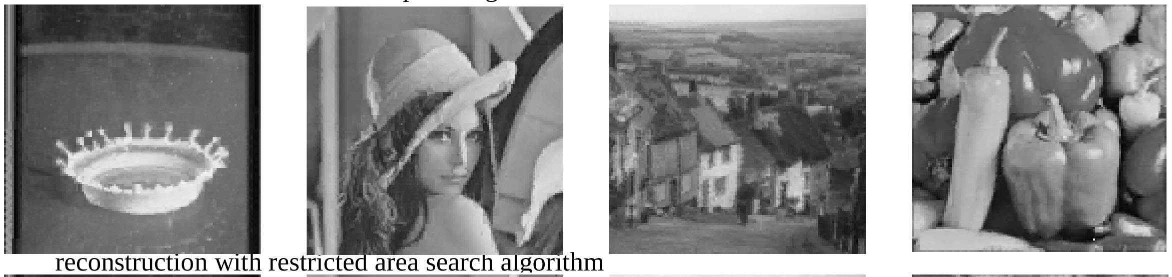
reconstruction with exhaustive search algorithm



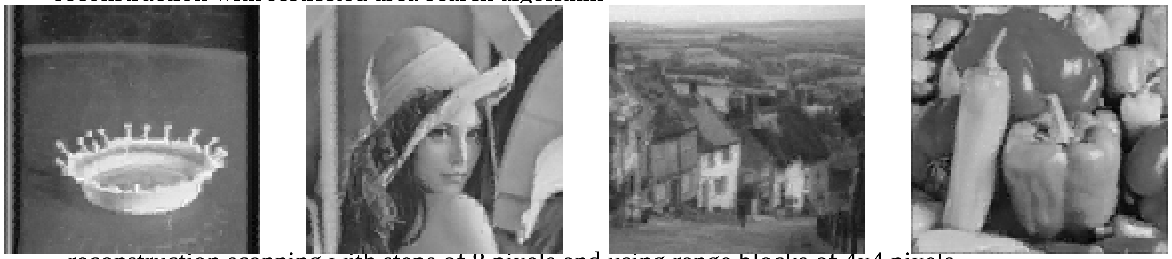
reconstruction with local search algorithm



reconstruction with look at same place algorithm

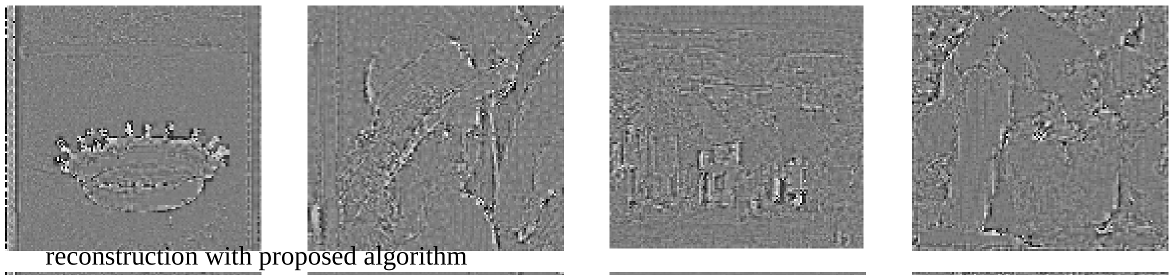


reconstruction with restricted area search algorithm

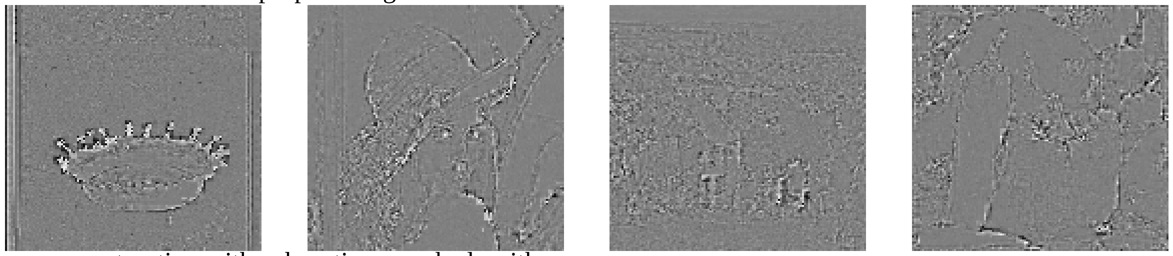


reconstruction scanning with steps of 8 pixels and using range blocks of 4x4 pixels

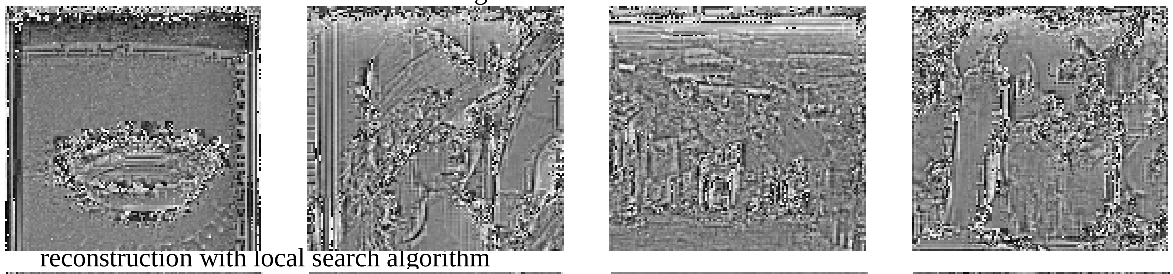
Fig. 4- Recontructed images using the proposed and other implemented algorithm.



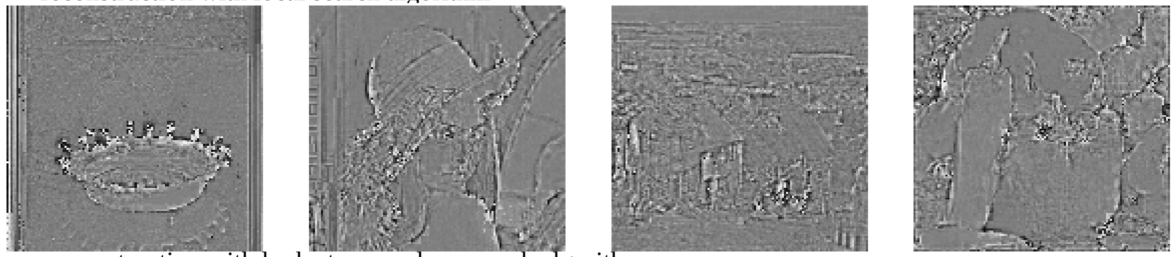
reconstruction with proposed algorithm



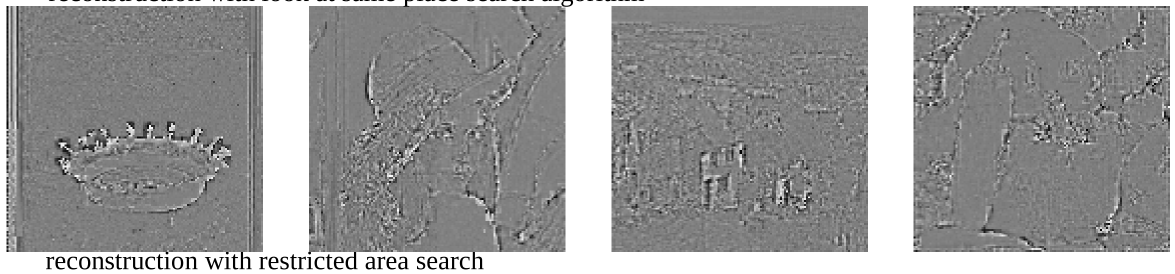
reconstruction with exhaustive search algorithm



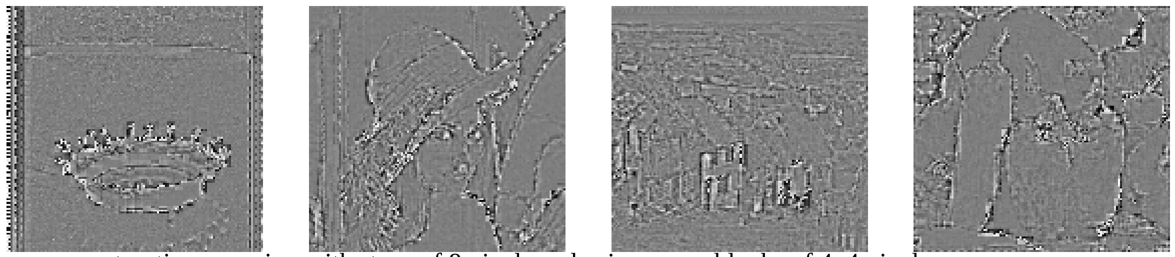
reconstruction with local search algorithm



reconstruction with look at same place search algorithm



reconstruction with restricted area search



reconstruction scanning with steps of 8 pixels and using range blocks of 4x4 pixels

Fig. 5- Error image using the proposed and other implemented algorithm. Each image is obtained by $\left(5\, e(x,y) + 255 \right) / 2$, where $e(x,y) = \left(f(x,y) - A(x,y) \right)$

The fidelity of an image after compression is an important aspect of lossy compression methods (lossless methods produce images that are an exact replica of the original). However, quality is not an easy issue to measure. Comparisons can be performed considering visual quality of the decompressed image in Figure 4. Figure 5 shows the error associated with the image generated by subtracting the differences between the original image given in Figure 3 and the decompressed images of Figure 4 and then scaling these differences between 0 and 255 so that they can be displayed as an image. Comparing the performance using these error images make it easier to see where the decompressed image has been slightly blurred, the extension of regions in which the high spatial frequencies have been removed, and the position where the fine detail of the image have been seriously lost.

For numerical comparative relation, one measure of a compressed image's fidelity is the root mean square error e_{rms} between the decompressed image *A(x,y)* and the original image *f(x,y)*, both of size *NxN*:

$$e_{rms} = \left(\Sigma_{x=i}^{N} \Sigma_{y=i}^{N} [\, e(x,y) \,]^{2} \right)^{0.5}$$

where $$e(x,y) = \left(f(x,y) - A(x,y) \right)$$

The smaller the e_{rms} is, the closer the compressed image is to the original. The root signal-to-noise ratio of the compressed image is then given by

$$SNR_{rms} = \left(\Sigma_{x=i}^{N} \Sigma_{y=i}^{N} [A(x,y)]^{2} / \Sigma_{x=i}^{N} \Sigma_{y=i}^{N} [\, e(x,y)]^{2} \right)^{0.5}$$

The closer the compressed image is to the original image, the higher the signal to noise ratio will be. The main difficulty in using e_{rms}, SNR and others as a measure of image quality is that in many instances these values do not match the quality perceived by the human visual system [Weeks Jr. (1996)]. The values of these features plus compression time and compression ratio on bits per pixel of the four image can be compared on Table 1.

The performance comparison uses five approaches. The first is the basic Barnsley and Hurd (1993), which performs an exhaustive search on all domain blocks for each range block.

The second approach, searches for t_i at the corresponding places of the domain and range blocks (Figure 6) and nowhere else. In this case the affine transformation is coded, and the vertical and horizontal position of the mapping are not included leading to a higher compression ratio [Monro et al. (1993)].

The third approach, named the local search algorithm, looks for the matching couple in 81 domain blocks. The search is only made inside an area containing 4 positions above, below, to the left, and to the right to the position of the range block (Figure 7).

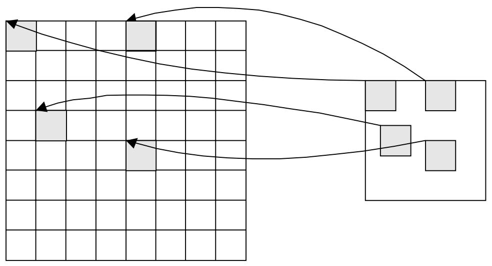


Fig. 6 - Search strategies of look in the same place approach.

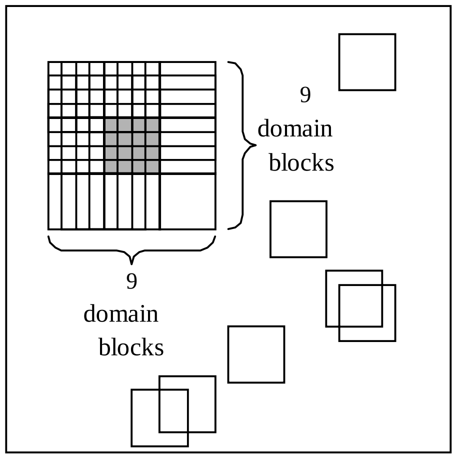


Fig. 7 - Domain blocks near the range block position used for best match examination on the here compared local search approach.

In the forth approach, the restricted area search algorithm, the main idea is to reduce the encoding time considering only domain blocks in the same quadrant of the range block (Kominek, 1995). For example, the image to be compressed may be sectioned into four quadrants. For a range block in the bottom left quadrant, one only needs to search for the domain blocks in that same quadrant (Figure 8). As a result, the search time is reduced by a factor of four over that of using an exhaustive search.

The fifth approach is the called light brute force, which restricts the domain blocks to be an integral multiple of a predefined step (Kominek, 1995). In our implementation of this approach, the range block under consideration have 4×4 pixels and a step of 8 pixels (Figure 9). The image quality of this approach suffers from the fact that not every domain block is considered, and the optimal pairing for a given range block may be missed.

The compression time is reduced using one of these search strategies, at the expense of some loss of image quality. The data of Table 1 and in Figures 4 and 5 were obtained using the same basic implementation running on the same platform. All images have the resolution 128×128 and 256 gray levels. The average row in Table 1 of each method is visualized in Figure 10 which helps in comparing different numerical aspects. The proposed modification to the standard fractal image compression method shows little loss of image quality. Only 2 approaches present errors smaller than the proposed. However, their errors are close to the error obtained with the proposed modification. Moreover, the compression times of these 2 approaches are at least twice the time of that of the new modification.

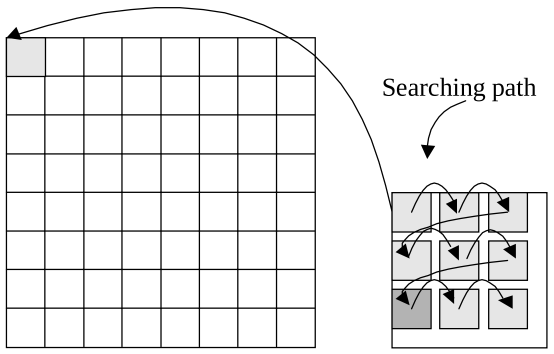


Fig. 8 - Search strategies of restricted area search algorithm.

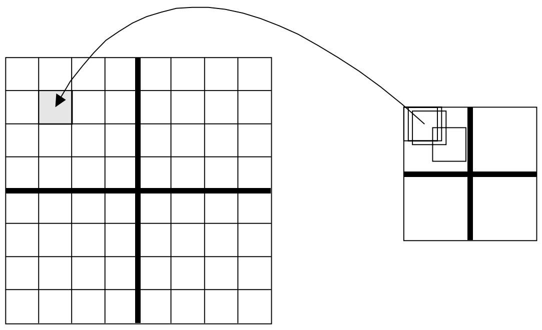


Fig. 9 - Search strategies of the called light brute force search algorithm

4. Conclusions

This work presents an original modification to the standard fractal image compression method. The approach uses the Local Fractal Dimension (LFD) of the image range blocks to select among the possible domain pools the most adequate to be used in the search for the best matching (between range and domain blocks). The domain pools are generated from the image to be compressed by the use of scale factors. The scale factors are related to the image LFD. A group of range blocks with a low LFD represents image regions with low complexity, where larger scale factor can be applied, which reduce the number of elements in the domain pool speeding up the search for the best domain/range pair. On the other hand, in image regions with large complexity, high LFD, the scale reduction must be adequate to permit detailed search in a more complex domain pool.

The efficiency of the proposed method is illustrated by four image samples. This modification speeds up the compression without a significant loss in the image quality or change in the compression ratio. The rms-error performance of 5 other approaches shows that only approaches with at least twice the compression time (restricted area search and exhaustive search) present slightly better quality results.

Acknowledgements

This research was supported by project FINEP/RECOPE SAGE #0626/96, FAPERJ and CNPq ref. 302649/87-5. The authors thank the anonymous reviewers for their suggestions, particularly: (1) division of the LFD into bands and comparison between range and domain blocks using adjacent bands and (2) to sort the domain block by their LFD and only compare for each range the domains with closest dimension.

References

F. R. Aquino, *Um Estudo sobre Metodologias de Codificação Fractal*, Dissertação de Mestrado, UFF-CAA, 1998.

M. F. Barnsley, *Fractals Everywhere*, 2nd Edition, Academic Press, New York, 1993.

M. F. Barnsley and L. Hurd, *Fractal Image Compression*. AK Peters, Wellesley, 1993.

N. Boulgouris and M. Strintziz, “Directional interpolation pyramidals for still image compression”, *Proceedings SIBGRAPI'98*; Rio de Janeiro, RJ (1998) 174—84.

Chwen-Jye Sze, Hong-Yuan Mark Liao, Kuo-Chin Fan, Ming-Yang Chern and Chen-Kou Tsao, “Fractal image coding system based on an adaptive side-coupling quadtree structures”, *Image and Vision Computing*, 14 (1996) 401—15.

A. Conci and F.R. Aquino, “Fractal image coding by multi-scale selection based on block complexity”, *Journal for Geometry and Graphics*, Heldermann Verlag, Germany, 3 (1999) 57—64.

A. Conci and C.F.J. Campos, “Can Fractal Dimension Be Used in Textile Imperfection Identification”, *Proceedings SIBGRAPI'95*, São Carlos, SP (1995) 127—33.

A. Conci and C.B. Proença, “A fractal image analysis system for fabric inspection based on a box-counting method", Computer Networks and ISDN Systems. Elsevier, 30 (1998) 1887—1895.

G. Franssens, M. Maziere, D. Fonteyn and D. Fussen, "Image compression based on a multipoint Taylor series representation", *Proceedings SIBGRAPI'98*; Rio de Janeiro, RJ (1998) 185—90.

Y. Fisher, *Fractal Image Compression: Theory and Application*, Springer-Verlag, New York, 1995.

J. C. Hart, "Fractal Image Compression and Recurrent Iterated Function Systems", *IEEE Computer Graphics and Applications*, July (1996) 25—40.

E.W. Jacobs, Y. Fisher and R. D Boss, "Image compression: a study of the iterated transform method", *Signal Processing*, 29 (1992) 251—63.

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

J. Kominek, "Advances In Fractal Compression For Multimedia Applications", *Internal Report CS95-28*, (ftp://links.uwterloo.ca:/pub/Fractals/Papers/Waterloo/kominek95c.ps.gz) University of Waterloo (1995).

C. S. Kubrusly, *An Introduction to Models and Decompositions in Operator Theory*, Birkhauser, Boston, 1997.

C. K. Lee and W. K. Lee, "Fast Fractal Image Block Coding Based on Local Variances", *IEEE Transactions on Image Processing*, 7, No. 6 (1998) 888—91.

N. Lu, *Fractal Imaging*, Academic Press, San Diego, 1997.

D. Monro, D. Wilson and J. Nicholls, "High speed image coding with Bath Fractal Transformation" *IEEE Int. Symposium on Multimedia Technologies*, (1993).

S. Morgan and A. Bouridane, "Application of Shape Recognition to Fractal Based Image Compression". *Proceedings IWSIP'96-3rd International Workshop in Signal/Image Processing* - Elsevier Sc. (1996) 665—8.

M. Nappi, G. Polese and G. Tortora, "FIRST: Fractal Indexing and Retrieval SysTem for Image Databases", *Image and Vision Computing*, 16 (1998) 1019—31.

S. Peleg, J. Naor, R. Hartley and D. Avnir, "Multiple resolution texture analysis and classification", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6 (1984) 518—23.

L. Rila. "Image coding using irregular subsampling and Delaunay triangulation", *Proceedings SIBGRAPI'98*; Rio de Janeiro, RJ (1998) 167—73.

N. Sarkar and B.B. Chaudhuri, "An efficient differential box-counting approach to compute fractal Dimension of Image", *IEEE Trans. Sys. Man and Cybernetics*, 24, No.1 (1994) 115—120.

D. Saupe and R. Hamzaoui. "A Review of Fractal Image Compression Literature", *Computer Graphics*, November. 28. (1994) 268—75.

A. R. Weeks. Jr. *Fundamentals of Electronics Image Processing*, IEEE & SPIE Press., 1996.

H. C. Yun, B.K. Guenter and R.M. Mersereau, "Lossless Compression of Computer-Generated Animation Frames", *ACM Trans. on Graphics*, 16, No. 4 (1997) 359—96.

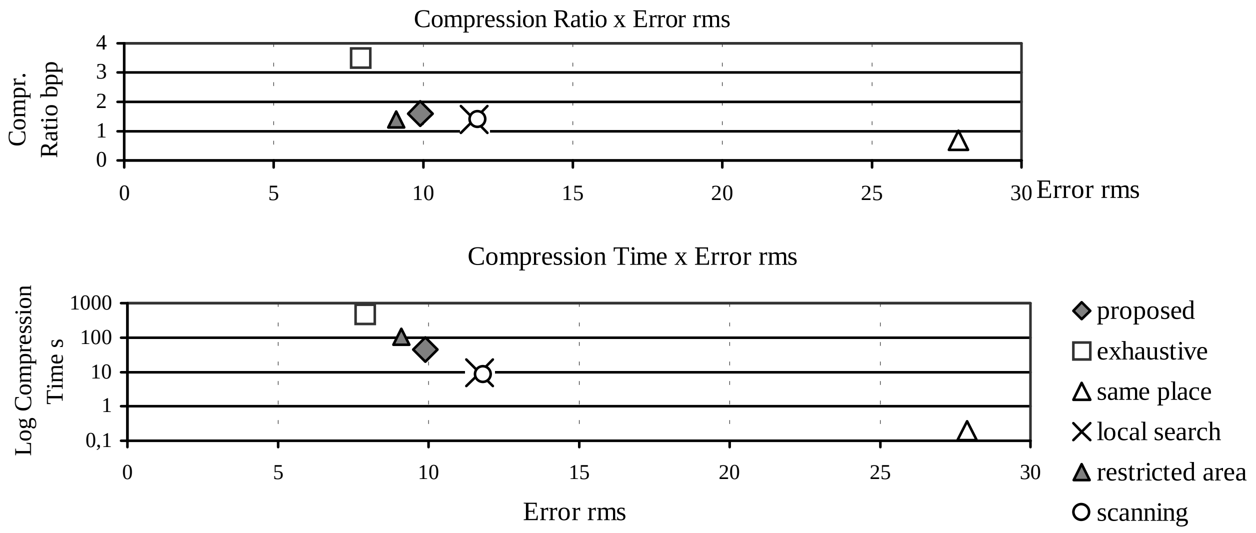


Fig. 10 - Visualization of the average performance of all implementation considering.