RATE - DISTORTION OPTIMIZED QUADTREE PARTITIONING METHOD

Macarie BREAZU

University "Lucian Blaga" of Sibiu, ROMANIA

e-mail: mac@cs.sibiu.ro

Abstract: The paper introduces a new method based on rate-distortion optimised quadtree partitioning to fractal image compression. The method starts with a complete quadtree splitting phase where the best corresponding domain for each range is searched and kept. Then, the corresponding tree is pruned according to a rate-distortion criterion. Experimental results prove that our method yields a better rate-distortion curve than the classic quadtree partitioning scheme. The classical (empirically chosen) tolerance flag is no longer needed and the stop condition is replaced by a more useful desired bit-rate (compression ratio) criterion.

1. INTRODUCTION

A major problem that researchers in fractal image compression face, beside reducing the encoding time, is image partitioning. Because of the huge searching space, the optimal partitioning problem for a desired bit-rate, cannot be practically solved. Until now, deterministically *hierarchical partitioning* (quadtree scheme [1], HV [2] and polygonal partitioning [3]) and *split-and-merge* methods ([4], Delaunay triangulations [5][6], quadrilateral [7], heuristic [8], evolutionary [9] and deterministic search [10],[11]) have emerged as solutions for the problem.

In order to comply with the spatial contraction of the fractal transform we consider the domains twice as large as the corresponding ranges (as usually). The spatial transform applied to a domain for matching the range size is the simple method of shrinking by pixel averaging. All the 8 isometries (4 rotations and 4 flips) that can be applied to a block are considered, which has the effect of correspondingly enlarging the domain pool to be searched in order to find the best match. For a range R and a domain D, we compute the scale s and offset o coefficients by minimizing the collage error as a function having s and o as parameters. The value obtained for s is then clamped to the [-1,1] interval in order to assure the contraction in luminance space as well. The collage error becomes, after applying an uniform quantization to the parameters s and o and yielding $\bar{s}$ and $\bar{o}$,

$$E(R,D) = \left\| R - (\bar{s}D + \bar{o}\mathbf{1}) \right\|^2$$

where **1** is a uniform image block with each pixel having unit intensity. The bit stream transmitted to the decoder contains the codebook index of the best corresponding domain and the quantized values $\bar{s}$ and $\bar{o}$.

2. FISHER's QUADTREE PARTITIONING METHOD

In the Fisher's quadtree method ([12], [13]) the image is partitioned into its four quadrants according to a tolerance flag: if the collage error of the best domain-range mapping is greater than the tolerance flag the block is divided in its 4 quadrants and the process continues recursively. Obviously, some limits for the minimum and maximum recursion depth are imposed.

In order to speed up the coding, blocks are classified into predefined classes as follows: a square block (range or domain) is subdivided into its four quadrants (specific for the quadtree method). For each quadrant the average pixel intensities A_i (i=1…4) and the corresponding variances V_i are computed. Applying the rotations and flips, the block can be ordered in one of the next three ways:

Major class 1: $A_1 \geq A_2 \geq A_3 \geq A_4$

Major class 2: $A_1 \geq A_2 \geq A_4 \geq A_3$

Major class 3: $A_1 \geq A_4 \geq A_2 \geq A_3$

Once the major class is established, there are 24 possible orderings of the variances that define 24 subclasses for each major class. In the original implementation of Fisher, the "fullclass_search" flag controls whether only one (of all 3) major class is searched, whereas the "subclass_search" flag controls whether only one of the 24 subclasses is searched for one major class. Certainly, extending the search to all major classes and to all subclasses improves the image quality but increases the encoding time.

The main disadvantage of this method is that it relies on a (empirically chosen) tolerance flag: a small value of this flag gives a high image quality and a long decoding time, whilst a great value of the flag does the opposite. Unfortunately, we do not know what value to choose in order to obtain some specific (imposed) compression ratio or image quality. Some results were reported in [15], where variable tolerance flags were used in order to improve the performance of the quadtree method.

3. THE NEW RATE-DISTORTION OPTIMIZED METHOD

We propose a new quadtree partitioning method that improves the rate-distortion curve and eliminates the need for any (empirically chosen) parameters. In a way, it continues the work presented in [11], being a particular partitioning case but with a rate-distortion merging criterion. On the other hand, our method is closely related to the well-known BFOS algorithm, also described in [14]. The proposed method falls in the category of split-and-merge methods and therefore has two phases:

1. **Splitting phase** (initialization phase): the image is quadtree partitioned recursively until the minimum accepted block size is achieved. At each (allowed) level of partitioning, for each range block the best domain is searched as in Fisher's scheme. A tree containing ranges in nodes is constructed, each tree node having 4 child trees corresponding to its 4 quadrants. For each range, its node stores the best domain-range mapping. Based on these, the following values are computed for each (non-leaf, merge-able) node of the tree, from leaf nodes to the root node:

$$\Delta E = E_B - \sum_{i=1}^{4} E_{S_i}$$

$$\Delta B = \sum_{i=1}^{4} N_{S_i} - N_B$$

representing the collage error increase and the bit-rate gain obtained by replacing the 4-child sub-trees (S_i) – based mapping with the single range block mapping. The following notations were used:

E_B - the collage error of the best domain-block mapping of the block B.

E_{S_i} - the collage error of the best mapping(s) of the child S_i sub-tree.

N_B – the number of bits needed to store the domain-block mapping of the block B.

N_{S_i} – the number of bits needed to store the child S_i sub-tree.

In all these cases, the collage error is defined as the square error of the domain-range mapping (different from Fisher's original implementation, where it is defined as the square root of the square error of the mapping).

2. **Merging phase** : In this phase, the previously constructed tree is pruned according to a rate-distortion criterion. The phase can be described as follows:

- repeat until the desired bit-rate is achieved
  - select the (merge-able) node which has the smallest value of the $\frac{\Delta E}{\Delta B}$ criterion
  - prune the sub-trees of the selected node

Tolerance flag	Time (sec)	Transformation bits	Partitioning bits	Percent of partitioning bits	Compresion ratio	PSNR (dB)
2	275	100937	1280	1.25 %	20.50	30.87
4	207	75867	1156	1.50 %	27.21	30.78
6	172	62293	1068	1.68 %	33.07	30.51
8	142	51917	1004	1.89 %	39.59	30.10
10	121	43869	928	2.06 %	46.76	29.46
12	106	38297	844	2.15 %	53.52	28.85
14	94	34217	792	2.25 %	59.83	27.28
16	81	29261	708	2.35 %	69.88	27.28
18	71	25607	680	2.58 %	79.65	26.77
20	61	21770	620	2.76 %	93.48	25.88
22	52	18762	572	2.95 %	108.23	25.27

Table 1 Results for classic Fisher quadtree partitioning method

Number of mergings	Merging time (sec)	Transformation bits	Partitioning bits	Percent of partitioning bits	Compresion ratio	PSNR (dB)
0000	0.00	102400	1280	1.23 %	20,21	30.87
0440	2.15	68557	1204	1.72 %	30,04	30.72
0668	2.97	51139	1112	2.12 %	40,10	30.24
0805	3.30	40645	1024	2.45 %	50,27	29.50
0895	3.52	33799	968	2.78 %	60,24	28.64
0960	3.63	28866	920	3.08 %	70,29	28.05
1009	3.68	25153	880	3.37 %	80,43	27.44
1046	3.74	22293	836	3.60 %	90,51	27.02
1076	3.79	20025	808	3.87 %	100,47	26.67
1101	3.85	18160	768	4.04 %	110,56	26.35
1121	3.85	16597	732	4.21 %	120,74	26.00

Table 2 Results for the new proposed quadtree partitioning method

- update the E_{S_i} and N_{S_i} values for all the ranges on the path to the root.

We notice the test for the desired bit-rate is easy to be done because the root node stores the value $\sum_{i=1}^{4} N_{S_i}$ which represents the number of bits needed to store the entire tree.

4. EXPERIMENTAL RESULTS

In order to test the proposed method we used a modified version of the Fisher's encoding program. All the tests hereinafter presented are performed in the following conditions:

- Standard 512*512 pixels, 8-bit grayscale Lena test image.
- The recursion quadtree level is chosen to allow range block sizes of 32*32, 16*16 and 8*8.
- Scale *s* and offset *o* parameters are uniform quantizated on 5 and 7 bits, respectively.
- Both positive and negative scalings are used.

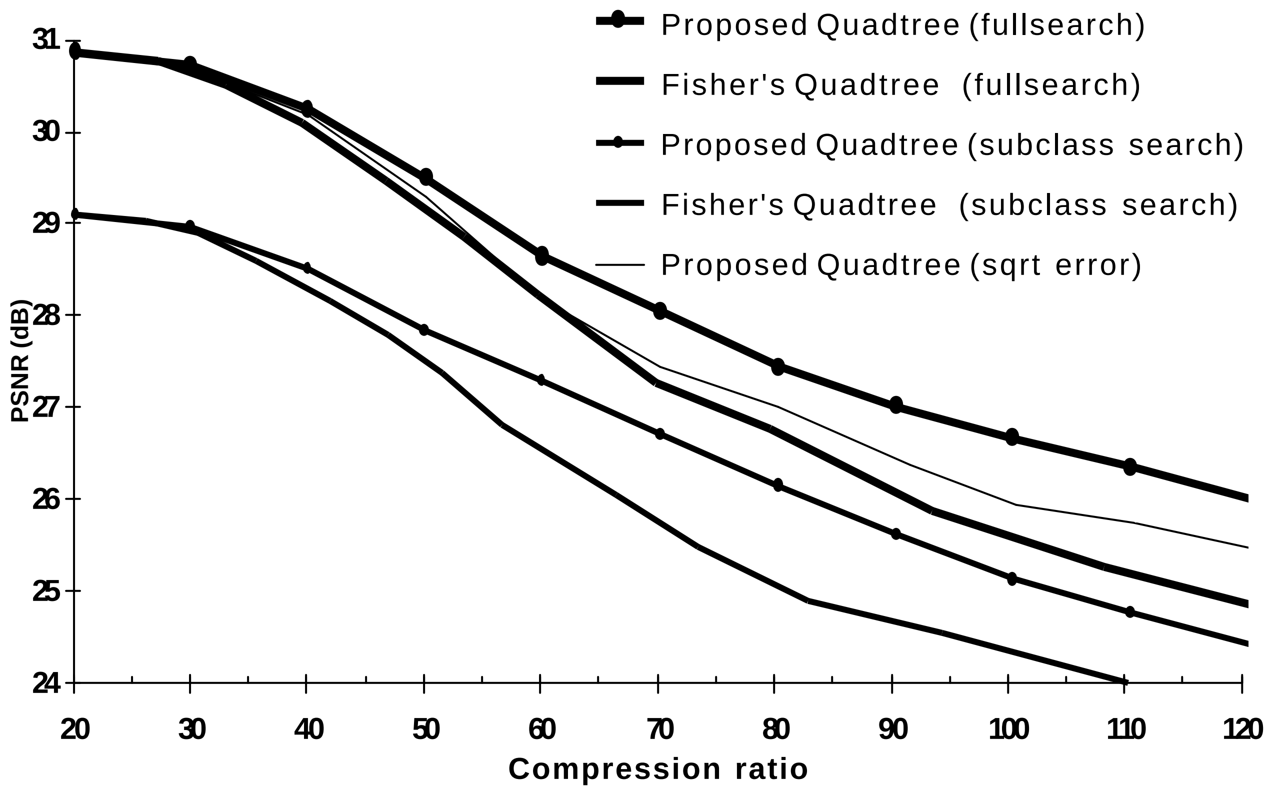
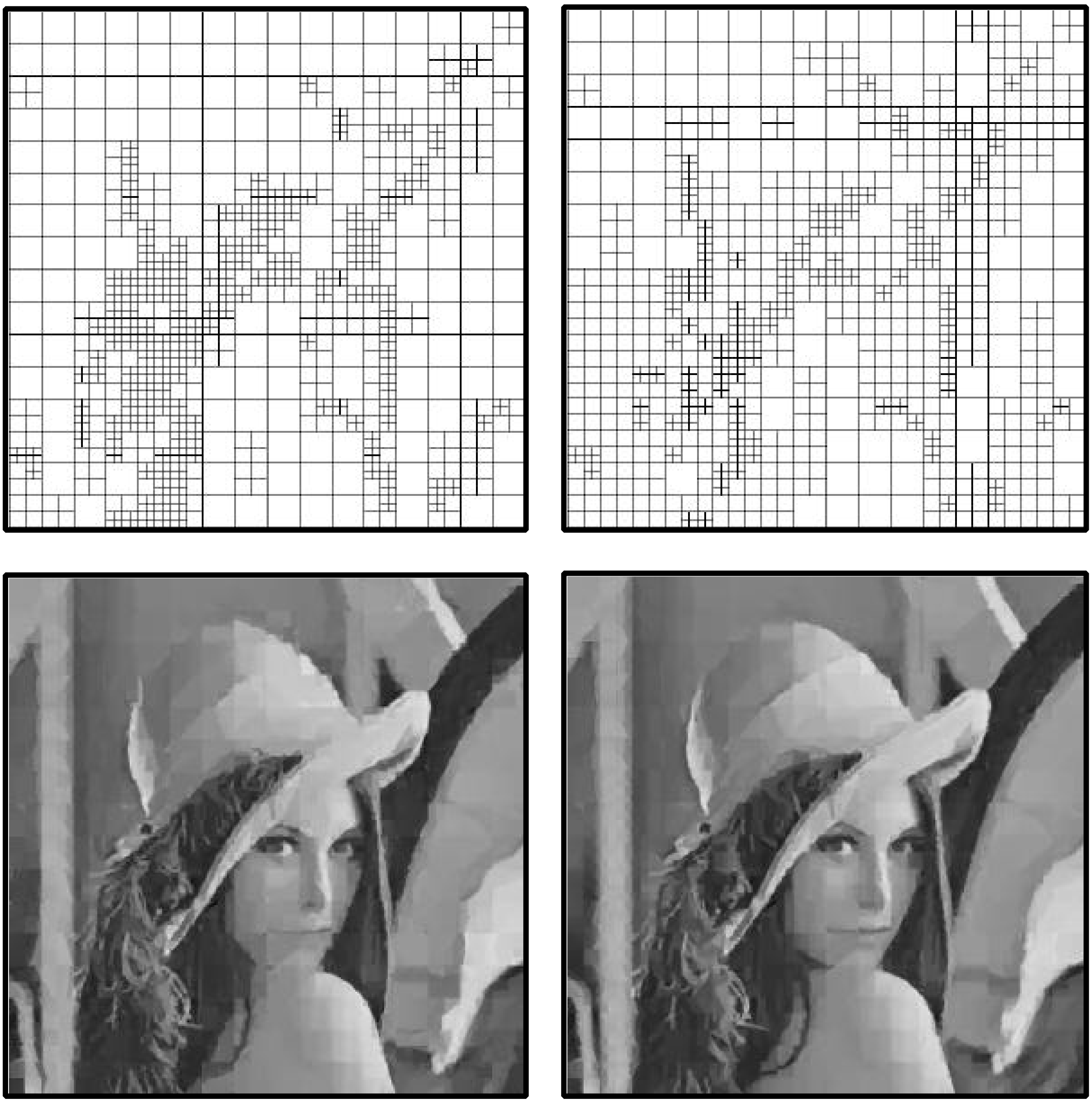


Figure 1 Rate-distortion curves

- The 8 well-known isometries of a block are taken into consideration.
- The domain position and isometry are not stored for uniform blocks (having s=0)
- Domains are twice as large as ranges, while the domain step is equal to the domain size (the default case in Fisher's program).

First we consider fullclass search and subclass search (corresponding to the flags fullclass_search=1 and subclass_search = 1). The results obtained using the Fisher's method are presented in Table 1 (here, the tolerance flag refers to the square root of the collage error, as in the original Fisher's program).

In Table 2 we present the results obtained for the new proposed method. Figure 1 presents results for both methods and proves the superiority of the proposed method. We notice the fraction of transformation bits increases a little (but still remains small), which can be accounted for by a better elimination of some useless range-domain mappings. In Figure 2 we present the image partitioning obtained using the two methods and the decoded image quality for both of them (corresponding to tolerance flag = 18 and number of mergings = 1009, respectively). The proposed method leads to a more gradual image partitioning and (even visible) better image quality.



CR=79,65 PSNR=26,77dB
Fisher's Quadtree Method

CR=80,43 PSNR=27,44dB
New Quadtree Method

Figure 2 Image partitioning and decoded image quality

Tests were repeated for the case of searching only one (corresponding) major class and subclass (i.e. subclass_search=0 and fullclass_search=0) and Figure 1 proves the proposed method maintains its superiority. The case of defining collage error as square root of the square error for the merging criterion is also tested and results presented in Figure 1 prove the criterion chosen in the paper is right.

Obviously, the total encoding time is greather for the proposed method than for the classical method but we don't consider this as a great disadvantage (the increase being acceptable). All the tests described in the paper are performed by running our modified quadtree partitioning compression program (maybe not best optimized) on an Intel Pentium 166 MMX based PC machine. Compatibility reasons with output file format used in [13] result in a bit-stream overhead of 39 configuration bits (not included in Tables 1 and 2).

5. CONCLUSIONS AND FUTURE WORK

A new quadtree based partitioning method is proposed. Experimentally, it is proven that the proposed method yields a better rate-distortion curve than the classical Fisher's quadtree partitioning scheme. Furthermore, there is no need for any empirically chosen parameter (i.e. the tolerance flag) and a simple and useful stop condition is readily available – the desired compression ratio. Certainly, suplimentary research should be done, especially regarding:

- Combining this method with other partitioning methods.
- The effect of entropy coding of the output bit-stream (separately coding each kind of bits).
- The effect of suboptimal searching methods for obtaining initial best domain-range mappings.

6. REFERENCES

[1] E. Jacobs, Y. Fisher, R.D. Boss, "Image compression: A study of the iterated transform method". *Signal Processing* 29, 1992, pp 251-263.

[2] Y. Fisher, S. Menlove, "Fractal encoding with HV partitions", in [12]

[3] E. Reusens, "Partitioning complexity issue for Iterated Functions Systems based image coding", *VIIth European Signal Processing Conference*, Vol. 1, Edinburgh, U.K., September 1994, pp 171-174

[4] M. Tanimoto, H. Ohyama, S. Katsuyama, T. Kimoto, T. Fujii, "A new fractal image coding employing blocks of variable shapes", *IEEE International Conference on Image Processing* (ICIP'96), Laussane, Sept. 1996

[5] F. Davoine, M. Antonini, J.-M. Chassery, M. Barlaud, "Fractal image compression based on Delaunay triangulation and vector quantization", *IEEE Transactions on Image Processing* Vol. 5, No. 2, February 1996, pp 338-346

[6] F. Davoine, E. Bertin, J.-M. Chassery, "From rigidity to adaptive tessellations for fractal image compression: comparative studies", *IEEE 8th Workshop on Image and Multidimensional Signal Processing*, Cannes, September 1993, pp. 56-57

[7] F. Davoine, J. Svensson, J.-M. Chassery, "A mixed triangular and quadrilateral partition for fractal image coding", *IEEE International Conference on Image Processing (ICIP'95)*, Washington, D.C., 1995

[8] L. Thomas, F. Deravi, "Region-based fractal image compression using heuristic search", *IEEE Transactions on Image Processing*, Vol. 4, No. 6, June 1995, pp 832-838

[9] D. Saupe, H. Hartenstein, "Evolutionary fractal image compression", *IEEE International Conference on Image Processing* (ICIP'96), Laussane, September 1996

[10] M. Ruhl, H. Hartenstein, D. Saupe, "*Adaptive Partitionings for Fractal Image Compression*", IEEE International Conference on Image Processing (ICIP'97), Santa Barbara, Oct. 1997.

[11] M. Breazu, G. Toderean, "*Region-based fractal image compression using deterministic search*", IEEE International Conference on Image Processing (ICIP'98), Chicago, Oct. 1998.

[12] Y. Fisher (ed.), *Fractal Image Compression - Theory and application*. New York, NY: Springer-Verlag, 1994.

[13] Enc.c / dec.c programs existing on Yuval Fisher's site **http://inls.uscd.edu/y/Fractals/**

[14] D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, D. Marini, "*Optimal hierarchical partitions for fractal image compression*", IEEE International Conference on Image Processing (ICIP-98), Chicago, Oct. 1998.

[15] Y.-C., Chang, B.-K., Shyu, J.-S., Wang, "*Region-based fractal image compression with quadtree segmentation*", ICASSP'97, Munich, 1997.