A new approach for improvement of fractal image encoding

Sofia Douda[1], Abdelhakim El Imrani[2], Abdallah Bagri[3]
[1] Department of Mathematics and Computer science, Faculty of Sciences and Techniques, Settat, Morocco
[2] Laboratory of Conception and Systems, Faculty of Sciences, Rabat, Morocco
[3] ENIC, Faculty of Sciences and Techniques, Settat, Morocco

***Abstract*—In this paper, we propose, in a first part, an approach to reduce the computational complexity of fractal image encoding by using the Shannon entropy (APENT). A speedup factor of 8 is obtained while image quality is still preserving. In a second part, we improve the APENT by using the AP2D approach that we have proposed in a previous study. We refer to this proposed approach as AP2D-ENT. The experimental results show that AP2D-ENT is effective in speeding up the encoding time without decreasing the image quality. Indeed, a speedup factor of 18 is reached for the test images with an increase of the compression ratio (CR) and a good image quality.**

Keywords- Fractal encoding, image compression, PIFS, complexity reduction, Shannon entropy.

I. INTRODUCTION

Fractal image compression (FIC) [1-5] is one of the recent methods of compression. It has generated much interest due to its promise of high compression ratios and to the advantage of having very fast decompression. Another advantage of FIC is its multi-resolution property. This method, which is based on the collage theorem [1], shows that it is possible to code fractals images by means of some contractive transformations defining an Iterated Function System (IFS). As natural signals do not often possess global self transformability, Jacquin [3] proposed to look for local or partial transformability what led to the first algorithm of compression by Local Iterated Function Systems (LIFS).

In FIC based on PIFS, a partitioning of the image is made where every elementary part (range block) is put in corresponding transformation with another part of a different scale (domain block) looked for in the image. The classical encoding method, i.e. full search (FS), is time consuming because for every range block, the corresponding block is looked for among all the domain blocks, i.e. domain pool. Several methods are proposed to reduce the time encoding of FIC and the most common approach is the classification scheme [6-10]. In this scheme, the domain and the range blocks are grouped in a number of classes according to their common characteristics. For each range block, comparison is made only for the domain blocks falling into its class. Fisher's classification method [6] constructed 72 classes for the image blocks according to the variance and intensity. In Wang et al. [10], four types of range blocks were defined based on the edge of the image. Methods based on reduction of the domain pool are also developed. Jacobs et al. uses skipping adjacent domain blocks [11] and Monro and Dudbridge localizes the domain pool relative to a given range block based on the assumption that domain blocks close to this range block are well suited to match it [12]. Saupe's Lean Domain Pool method discards a fraction of domain blocks having small variance [13]. Other approaches focused on improvements of the FIC by tree structure search methods [14, 15], parallel search methods [16, 17] or by using two domain pools in two steps (AP2D approach) [18]. Also, the spatial correlation in both the domain pool and the range pool is added to improve FIC as developed by Truong et al. [19]. Tong [20] proposes an adaptive search algorithm based on the standard deviation (STD). During the step of search, the comparison is made only if the STD difference between the range block and domain block does not exceed a fixed threshold. The characteristic of these methods is that they differ by the time reduction, the image quality and the compression ratio. In the present work, we present, in a first part, a new approach to reduce the encoding time of FIC by using Shannon entropy (APENT) and in a second part we improve APENT by using AP2D approach. We refer to this method as AP2D-ENT. Experimental results show a higher time reduction without diminishing the image quality and with preserving the compression ratio.

II. FRACTAL IMAGE COMPRESSION

A. *Review of basic scheme*

The aim of FIC, is to find a mapping W in the image space so that the fixed point of this mapping exists, is unique, and is close as possible to the image I we want to encode. W is itself composed of N different block-wise mapping w_i, i=1,…, N. Each mapping w_i maps a domain block onto a smaller range block of size BxB. A search domain pool is created from the image taking all the square blocks (domain blocks) D_j, $\{D_1, D_2, …, D_q\}$ of size 2Bx2B, with integer step P in horizontal or vertical directions. To enlarge the variation, each domain is expanded with the eight basic square block orientations by rotating 90 degrees clockwise the original and the mirror domain block. The mapping w_i consist in two parts: a spatial part, translating from the domain position in the image to the range position and scaling the domain size down to the range size, and a massic part, modifying the pixels in the block. The massic part is itself composed of two transformations: a scaling operation, multiplying all the pixel values in the block by a coefficient s, and an offset operation, adding a constant coefficient o to each pixel value. A local transformation w_i is defined by :

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (1)$$

with pixel coordinates (x, y) and grey level z. The parameters (a_i, b_i, c_i, d_i) represent simple geometric transformations. The parameters (e_i, f_i) are the spatial translation vector.

The encoding process consists in finding for each range R_i, the best domain in the domain pool: the one that gives the least mean square error (MSE) when modified the scaling and offset operations. Of course, the coefficients s and o are computed so that they minimize the error, using a simple regression formula.

Given two squares containing n pixel intensities, $d_1,\ldots,d_n$ (from D_j) and $r_1,\ldots,r_n$ (from R_i). We can seek s and o to minimize the error between R_i and D_j quantity:

$$E(R_i, D_j) = \sum_{i=1}^{n} (sd_j + o - r_i)^2 \quad (2)$$

This will give us a contrast and brightness setting that makes the affine transformed d_j values have the least squared distance from the r_i values. The minimum of $E(R_i, D_j)$ occurs when the partial derivatives [4] with respect to s and o are zero, which occurs when:

$$s = \frac{n\sum_{i=1}^{n} d_i r_i - \sum_{i=1}^{n} d_i \sum_{i=1}^{n} r_i}{n\sum_{i=1}^{n} d_i^2 - (\sum_{i=1}^{n} d_i)^2} \quad (3)$$

and

$$o = \frac{1}{n}(\sum_{i=1}^{n} r_i - s\sum_{i=1}^{n} d_i) \quad (4)$$

The parameters o and s are the transformation parameters, corresponding to the current treated range bloc R_i. These parameters are quantified and coded to be used when decompressing the image. The value of $E(R_i, D_j)$ is the distance between blocs. It is compared to the used tolerance level e_c to decide if a given transformation is accepted or not.

The parameters that need to be stored are s, o, index of the best matching domain, and rotation index for each range block. The decoding simply consists in iterating the mapping W from any initial image and the result will be an attractor resembling the original image at the chosen resolution.

B. Basic fractal image encoding algorithm

The main step of the encoding algorithm of FIC by FS based on quadtree [6] is described as follows:

- Set Maxsize the maximum size of range blocks.
- Set Minsize the minimum size of range blocks.
- Choose a tolerance level e_c.
- Construct a domain pool D(P) where P is the step.
- Partition the image by a collection of range blocks R_i of size Maxsize and mark them as uncovered blocks.
- For every uncovered range block R_i :
- For every D_j of the domain pool D(P):
- Compute the coefficients s and o of the transformation w_i such as the $r_{ij}=d_{L2}(w_i(D_j), R_i)$ is minimal.
- Retain the reference of D_j and the coefficients s and o such as the distance r_{ij} corresponding is minimal.
- If the distance r_{ij} is lower than the threshold e_c:
    - Encode the reference of the domain block and the coefficients of the transformation.
- Else if the size of R_i is still small than Minsize:
    - Subdivide R_i in four sub-range blocks and add them to the uncovered range blocks.

III. THE PROPOSED IMPROVEMENT OF FIC

A. Shannon entropy

The Shannon entropy provides a way to estimate the average minimum number of bits needed to encode a string of symbols based on the frequency of the symbols.

Let be X= {x_1, x_2, …, x_n} a set of events with the probability of occurrence of each event $P(x_i)=P_i$. These probabilities, P={P_1, P_2, …, P_n}, are such that each $P_i \geq 0$, and

$$\sum_{i=1}^{n} P_i = 1 \quad (5)$$

The Shannon entropy takes the form:

$$H(P_1, P_2, \ldots, P_n) = H(X) = -\sum_{i=1}^{n} P_i \log P_i \quad (6)$$

The function H has the following lower and upper limits:

$$0 = H(1, 0, \ldots, 0) \leq H(P_1, P_2, \ldots, P_n) \leq H(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) = \log n \quad (7)$$

B. The proposed approach based on entropy

The most computationally intensive part of the FIC process is the search step. One way to decrease encoding time is to decrease the number of comparisons between range and domain blocks. The proposed approach APENT reduces the encoding time of FIC by using the value of entropy of range and domain blocks. Each range block is compared only with domain blocks having their entropy close to that of the range block. Indeed, for each range block R_i we search a domain block D_j such:

$$R_i \approx W_i(D_j) \quad (8)$$

From (8), we can write:

$$\text{entropy}(R_i) \approx \text{entropy}(W_i(D_j)) \quad (9)$$

The mapping W_i can be written as:

$$W_i(D_j) = W_i^0 \circ W_i^1 \circ W_i^2(D_j) \quad (10)$$

w_i^1 and w_i^0 are respectively a massic part and an isometric transformation. The transformation W_i^2 scales the domain size to the range size. Let us consider $W_i^2(D_j) = \tilde{D}_j$, where $\tilde{D}_j$ is the decimated domain block of D_j.

The massic part w_i^1 is itself composed of two transformations: a scaling operation, multiplying all the pixel values in the block by a coefficient s_i and an offset operation adding a constant o_i. We can write:

$$w_i^1(\tilde{D}_j) = s_i \tilde{D}_j + o_i \quad (11)$$

where s_i and o_i are the scaling and the offset coefficients. From (9), we can write

$$\text{entropy}(R_i) \approx \text{entropy}(w_i^0(s_i \tilde{D}_j + o_i)) \quad (12)$$

The translation and the isometric transformation did not change the entropy. Then (12) become:

$$\text{entropy}(R_i) \approx \text{entropy}(s_i \tilde{D}_j) \quad (13)$$

There exist an ε such as:

$$\left| \text{entropy}(R_i) - \text{entropy}(s_i \tilde{D}_j) \right| \le \varepsilon \quad (14)$$

Now because that $|s_i| < 1$ and after simplification we have:

$$\left| \text{entropy}(R_i) - \text{entropy}(\tilde{D}_j) \right| \le \varepsilon \quad (15)$$

Thus for every range block R_i, comparison is made only with domain blocks D_j that satisfies the condition (15). The value of parameter ε determines the set of domain block that participate to the search of the best block that matches a given range block. If ε is small, the encoding time is reduced but the image quality is diminished.

C. The proposed AP2D-ENT

To obtain more improvement of FIC, we propose to improve APENT by using the AP2D approach. We refer to this new scheme as AP2D-ENT. As we have mentioned in [19], AP2D may be combined to another method to reduce more the computation time. The advantage of AP2D is that it reduces the time encoding without a loss of CR and with a slight decrease of PSNR. In AP2D-ENT, we use two domain pools instead of one domain pool and the encoding of an image is made in two steps. In the two steps of encoding, each range block is only compared to the domain blocks satisfying equation (15).

D. Algorithm of AP2D-ENT

The algorithm of AP2D-ENT is described bellow, where Maxsize, Minsize and e_c are definde in the previous algorithm:

Step 1:

- Set a value for the step P and construct the domain pool D(2P).
- Choose a value of the parameter ε.
- Partition the image by a collection of range blocks R_i of size Maxsize and mark them as uncovered blocks.
- For every uncovered range block R_i :
- For every D_j of the domain pool D(2P):
- If | entropy(R_i)-entropy($\tilde{D}_j$) | ≤ ε :
  - Compute the coefficients s and o such as the $r_{ij}=d_{L2}(wi(D_j), R_i)$ is minimal.
  - Retain D_j and the coefficients s and o such as the distance r_{ij} corresponding is minimal.
  - If the distance r_{ij} is lower than the threshold e_c:
    - Encode the reference of the best D_j and the coefficients of the transformation w_i.
  - Else if the size of R_i is still small than Minsize:
  - Subdivide R_i in four sub-range blocks and add them to the uncovered range blocks.
  - Save the reference of the best domain block and the corresponding coefficients s and o.

Step 2:

- Construct the second domain pool D(P), discard the domain blocks belonging to D(2P) and set D'(P)= D(P)-D(2P).
- If R_i is badly approached by the first approximation:
  - Set $d'_{ij} = \min d_{L2}(w_i(D_j), R_i)$
  - Search a D_j in D'(P) that verify:

    | entropy(R_i)-entropy($\tilde{D}_j$) |≤ ε and

    $d_{ij}=d_{L2}(w_i(D_j),R_i)$ is minimal
  - If d_{ij} <d'_{ij} then replace the reference of the best domain block (found by the first approximation) by that of the new best domain block.

IV. Experimental results

The different tests are performed on three 256x256 images, represented in figure 1, with 8 bpp on PC with Intel Pentium Dual 2.16 Ghz processor and 2 GO of RAM. The partition quadtree is adopted for FIC. The encoding time is measured in seconds. The quality of image is measured by peak signal to noise ratio (PSNR) and the rate of compression is represented by the compression ratio (CR), i.e. the size of the original image divided by the size of the compressed image. The speedup factor (SF) of a particular method can be defined as the ratio of the time taken in full search to that of the said method, i.e.,

$$SF = \frac{\text{Time taken in full search}}{\text{Time taken in a particular method}}$$

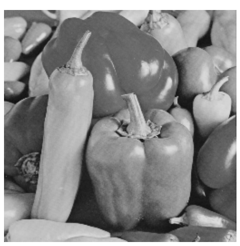
The PSNR of two images A and B of sizes n x n is defined as:

$$PSNR = 10x\log(\frac{255^2}{MSE}) \quad (16)$$

where

$$MSE = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(A(i,j) - B(i,j))^2}{n \times n} \quad (17)$$

(a)



(b)

(c)

Figure 1. Images of size 256 x 256 : Lena (a), Peppers (b) and San256 (c).

A. APENT

Table 1 and 2 gives the encoding time, the compression ratio and image qualities measured on the three test images for different values of ε by APENT. The last row of the table shows the results of FS. Figure 2 shows that the encoding time increases linearly with ε. For high values of ε (ε > 0.6), there is a slight reduction in time which become close to that of FS because the number of the domain blocks selected increased. The highest decrease of the PSNR (0.86, 1.09 and 1.66 for Lena, Peppers and baboon respectively) correspond to a speedup factor superior to 8.

For comparison, FS reach a PSNR of 30.92 dB with a required time of 20.80 seconds for Lena image whereas in APENT, we obtain a PSNR of 30.34 at encoding time of 3.56 seconds. This represents a speedup factor of 6.38 with a drop of PSNR of 0.58 dB.

TABLE I. EFFECT OF VARYING ε ON TIME, CR AND PSNR FOR LENA AND PEPPERS IMAGES.

ε	Lena				Peppers			
	Time	***CR***	***PSNR***	***SF***	***Time***	***CR***	***PSNR***	***SF***
0.02	2.57	8.96	30.06	8.09	2.27	9.29	30.82	8.93
0.04	2.69	9.26	30.15	7.73	2.47	9.57	30.94	8.21
0.06	3.31	9.43	30.26	6.28	2.85	9.71	31.16	7.12
0.08	3.56	9.50	30.34	5.84	3.18	9.91	31.18	6.38
0.1	3.79	9.55	30.38	5.49	3.34	10.07	31.35	6.07
0.2	6.34	9.90	30.71	3.28	5.63	10.29	31.53	3.60
0.4	10.27	10.24	30.85	2.03	9.02	10.70	31.67	2.25
0.6	12.99	10.41	30.85	1.60	11.50	10.90	31.77	1.76
0.8	15.45	10.44	30.87	1.35	14.00	10.94	31.84	1.45
1.0	17.36	10.44	30.89	1.20	15.41	10.98	31.87	1.32
1.2	18.42	10.44	30.9	1.13	16.53	10.98	31.87	1.23
1.4	19.28	10.46	30.91	1.08	17.41	10.98	31.89	1.16
1.6	20.06	10.46	30.91	1.04	18.60	10.98	31.90	1.09
1.8	20.56	10.46	30.91	1.01	20.08	10.98	31.90	1.01
2.0	20.75	10.46	30.92	1.00	20.26	10.98	31.91	1.00
FS	20.80	10.46	30.92	1.00	20.28	10.98	31.91	1.00

TABLE II. EFFECT OF VARYING ε ON TIME, CR AND PSNR FOR SAN256 IMAGE.

ε	San256			
	Time	**CR**	**PSNR**	**SF**
0.02	2.72	7.48	28.47	9.14
0.04	2.96	7.61	28.58	8.40
0.06	3.46	7.65	28.89	7.19
0.08	3.78	7.69	29.01	6.58
0.1	3.97	7.70	29.01	6.26
0.2	6.66	7.82	29.70	3.73
0.4	10.48	8.02	29.96	2.37
0.6	13.31	8.14	30.06	1.87
0.8	15.62	8.23	30.10	1.59
1.0	17.53	8.24	30.11	1.42
1.2	19.47	8.24	30.12	1.28
1.4	20.97	8.25	30.13	1.19
1.6	22.24	8.25	30.13	1.12
1.8	23.11	8.25	30.13	1.08
2.0	23.90	8.25	30.13	1.04
FS	24.87	8.25	30.13	1.00

The proposed approach did not cause a loss of image quality. For $\varepsilon \geq 0.2$ the maximal drops of the PSNR are 0.21 dB for Lena, 0.38 dB for Peppers and 0.43 dB for San256. We note that APENT produces very little influence on the CR especially when $\varepsilon \geq 0.2$. It appears that the best compromise between the time encoding, CR and PSNR for the test images is obtained when ε=0.1. In the case where ε =0.02, speedup factors for Lena, Peppers and San256 are 8.09, 8.93 and 9.14 respectively with a decrease of CR of 1.5, 1.69 and 0.77 respectively. The decrease of CR could be explained by the fact that some larger range blocks could be covered well by some domain blocks which are excluded from the domain pool because they don't satisfy the condition (16). Therefore these large range blocks are subdivided in four quadrants resulting in a decrease of CR.

For visual comparison, figures (3), (4) and (4) shows examples of reconstructed images encoded using FS and APENT. The advantage of the proposed approach is seen clearly on the three images.

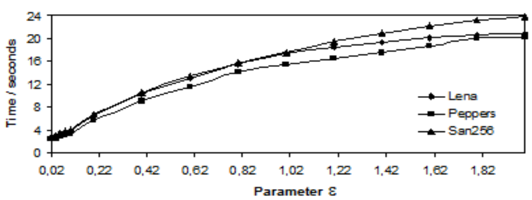


Figure 2. Effect of parameter ε on encoding time for the three images.

(a) Encoding time: 20.80 s, Quality: 30.92 dB

(b) Encoding time: 3.56 s, PSNR: 30.34 dB

(c) Encoding time: 2.57 s, PSNR: 30.06 dB

Figure 3. Lena image encoded by exhaustive search (a) and by APENT (b and c).

(a) Encoding time: 20.28 s, PSNR: 31.91 dB

(b) Encoding time: 3.18 s, PSNR: 31.18 dB

(c) Encoding time: 2.27 s, PSNR: 30.82

Figure 4. Peppers image encoded by exhaustive search (a) and by the proposed approach (b and c).

(a) Encoding time: 24.87 s, PSNR: 30.13dB

(b) Encoding time: 3.78 s, PSNR: 29.01dB

(c) Encoding time: 2.72 s, PSNR: 28.47dB

Figure 5. San256 image encoded by exhaustive search (a) and by the proposed approach (b and c)

B. *AP2D-ENT*

The results of AP2D-ENT are shown in Table 3, 4 and 5 for the test images. The results obtained show that AP2D-ENT is efficient in time reduction and CR without decreasing the quality of image (figure 6). Indeed, for Lena image, the speedup factor reaches 19.62 with a decrease of PSNR of 1.31 and with a diminution of CR of 0.93 in comparison to FS. For a speedup factor of 5.62, we note a decrease of PSNR of 0.48 and an increase of CR of 0.22. This enhancement of CR is caused by AP2D. The results also show that for a speedup factor inferior to 6, we obtain a slight decrease of PSNR and an increase of CR. For the lowest speedup factor (≥3) obtained by AP2D-ENT, we obtain the performance of FS with the advantage of an increase of CR. Consequently, AP2D-ENT provides the best results. For visual comparison figure (6) shows examples of reconstructed images encoded by AP2D-ENT.

TABLE III. THE RESULTS OF AP2D-ENT FOR DIFFERENT VALUES OF ε OF LENA IMAGE.

ε	AP2P-ENT			SF
	Time	*CR*	*PSNR*	
0.02	0.84	8.84	28.96	24.76
0.04	0.94	9.26	28.54	22.13
0.06	1.06	9.53	29.61	19.62
0.08	1.19	9.62	29.67	17.48
0.1	1.23	9.76	29.69	16.91
0.2	1.95	10.17	30.14	10.67
0.4	2.98	10.50	30.42	6.98
0.6	3.70	10.68	30.44	5.62
0.8	4.38	10.70	30.47	4.75
1	4.88	10.70	30.51	4.26
1.2	5.26	10.70	30.52	3.95
1.4	5.57	10.70	30.52	3.73
1.6	5.69	10.73	30.52	3.66
1.8	5.80	10.73	30.52	3.59
2	5.90	10.73	30.52	3.53

TABLE IV. THE RESULTS OF AP2D-ENT FOR DIFFERENT VALUES OF ε OF PEPPERS IMAGE.

ε	AP2P-ENT			SF
	Time	CR	PSNR	
0.02	0.86	9.23	29.10	23.58
0.04	0.94	9.64	29.71	21.57
0.06	1.00	9.88	29.93	20.28
0.08	1.12	9.99	30.59	18.11
0.1	1.19	10.12	30.60	17.04
0.2	1.90	10.51	31.05	10.67
0.4	2.92	10.81	31.37	6.95
0.6	3.73	11.03	31.42	5.44
0.8	4.34	11.12	31.49	4.67
1	4.71	11.12	31.50	4.31
1.2	5.13	11.17	31.50	3.95
1.4	5.35	11.17	31.51	3.79
1.6	5.51	11.17	31.52	3.68
1.8	5.59	11.17	31.53	3.63
2	5.68	11.17	31.53	3.57

TABLE V. THE RESULTS OF AP2D-ENT FOR DIFFERENT VALUES OF ε OF SAN256 IMAGE.

ε	AP2D-ENT			SF
	Time	*CR*	*PSNR*	
0.02	1.04	7.66	26.30	23.91
0.04	1.09	7.89	27.46	22.82
0.06	1.20	8.02	27.89	20.73
0.08	1.31	8.12	28.05	18.98
0.1	1.34	8.14	28.07	18.56
0.2	2.17	8.22	28.81	11.46
0.4	3.21	8.42	29.19	7.75
0.6	3.98	8.58	29.32	6.25
0.8	4.59	8.69	29.39	5.42
1.0	5.12	8.73	29.42	4.86
1.2	5.57	8.73	29.43	4.46
1.4	6.04	8.75	29.43	4.12
1.6	6.37	8.75	29.44	3.90
1.8	6.66	8.75	29.44	3.73
2.0	6.83	8.75	29.44	3.64

(a) Encoding time: 1.06 s, Quality: 29.61dB

(a) Encoding time: 1.12 s, Quality: 30.59 dB

Figure 6. Reconstructed images by AP2D-ENT for images Lena (a) and Peppers (b)

Table 6, 7 and 8 summarize the results of comparison between APENT, AP2D-ENT and Tong's STD method for the test images. It appears clearly that AP2D-ENT reach a high speedup factor with preserving the image quality and with an enhancement of CR.

TABLE VI. LENA IMAGE EXPERIMENTAL RESULTS.

	Time	CR	PSNR	SF
STD	3.84	9.23	29.36	5.42
APENT	2,57	8,96	30,06	8.09
AP2D-ENT	1.06	9.53	29.61	19.62

TABLE VII. PEPPERS IMAGE EXPERIMENTAL RESULTS.

	Time	CR	PSNR	SF
STD	3.45	9.08	29.39	5.88
APENT	2.27	9.29	30.82	8.93
AP2D-ENT	1.00	9.88	29.93	20.28

TABLE VIII. SAN256 IMAGE EXPERIMENTAL RESULTS.

	Time	CR	PSNR	SF
STD	4.04	7.7	26.05	6.16
APENT	2.72	7.48	28.47	9.14
AP2D-ENT	1.20	8.02	27.89	20.73

V. CONCLUSION

In this paper, we present an approach based on the Shannon entropy (APENT) to improve FIC in a first part. Experimental results show that a speedup factor of 8 is obtained for the test images without remarkable deterioration of image quality. We also propose to enhance APENT by using the AP2D in a second part. We obtain a high speedup factor with preserving image quality and with an increase of the compression ratio.

REFERENCES

[1] M. F. Barnsley and A. D. Sloan, "A better way to compress images", BYTE magazine, pp. 215-223, 1988.

[2] M. F. Barnsley, "Fractal every where". New-york: Academic Press, California, 1988.

[3] A. E. Jacquin, "A fractal theory of iterated Markov operators on spaces of measures with applications to digital image coding", PhD Thesis, Georgia Institute of Technology, 1989.

[4] A. E. Jacquin, "A novel fractal block coding technique for digital image", IEEE Int.Conf. on ASSP, ICASSP-90, pp. 2225-2228, 1990.

[5] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations", IEEE Trans. on Image Processing, Vol. 1, pp.18-30, January 1992.

[6] Y. Fisher, "Fractal Image Compression: Theory and Application", Springer-verlag, New York, 1994.

[7] D. J. Duh, J. H. Jeng and S. Y. Chen, "DCT based simple classification scheme for fractal image compression", Image and vision computing, Vol. 23, pp. 1115-1121, 2005.

[8] X. Wu, D. J. Jackson and H. Chen, "A fast fractal image encoding method based on intelligent search of standad deviation", Computers and Electrical Engineering, Vol. 31, pp. 402-421, 2005.

[9] T. Kovacs, "A fast classification based method for fractal image encoding", Image and Vision Computing, Vol. 26, pp. 1129-1136, 2008.

[10] Z. Wang, D. Zhang and Y. Yu, "Hybrid image coding based on partial fractal mapping", Signal Process: Image Commun., Vol. 15. pp. 767-779, 2000.

[11] E. W. Jacobs, Y. Fisher and R. D. Boss, "Image compression: A study of iterated transform method", Signal process, Vol. 29, pp. 251-263, 1992.

[12] D. M. Monro and F. Dudbridge, "Approximation of image blocks", In Proc. Int. Conf. Acoustics, Speed. Signal Processing, Vol. 3, pp. 4585-4588, 1992.

[13] D. Saupe, "Lean domain pools for fractal image compression", Journal of Electronic Imaging, Vol. 8, pp. 98-103, 1999.

[14] B. Bani-Eqbal, "Enhancing the speed of fractal image compression, Optical Engineering", Vol. 34, No. 6, pp. 1705-1710, 1995.

[15] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding still images", in Proc. EOS/SPIE Visual Communications PACS Medical Applications, Vol. 1977, pp. 397-408, 1993.

[16] C. Hufnagel and A. Uhl, "Algorithms for fractal image compression on massively parallel SIMD arrays", Real-Time Imaging, Vol. 6, pp. 267-281, 2000.

[17] D. Vidya, R. Parthasarathy, T. C. Bina and N. G. Swaroopa, "Architecture for fractal image compression". J. Syst. Archit., Vol. 46, pp. 1275-1291, 2000.

[18] S. Douda, A. El Imrani and M. Limouri, "Une nouvelle approche d'accélération du codage fractal d'images", ARIMA, Vol. 11, pp. 97-114, 2009.

[19] T. K. Troung, C. M. Kung, J. H. Jeng and M. L. Hsieh, "Fast fractal image compression using spatial correlation", Chaos Solitons & Fractals, Vol. 22, pp. 1071-1076, 2004.

[20] C. S. Tong, M. Pi, "Fast fractal image encoding based on adaptive search", IEEE Trans Image Process, Vol. 10, pp.1269-1277, 2001.