Electronic Letters on Computer Vision and Image Analysis 10(1):11-23, 2011

A reduced domain pool based on DCT for a fast fractal image encoding

Sofia Douda[1,2], Abdallah Bagri[2], Amer Abdelhakim El Imrani[3]
[1]Département de Mathématiques et Informatique, Faculté des Sciences et Techniques, Settat, Morocco
[2] ENIC, LBN, Faculté des Sciences et Techniques, Settat, Morocco
[3] LCS, Faculté des Sciences, Rabat, Morocco

Received 11th Nov 2010; accepted 09th Oct 2011

Abstract

Fractal image compression is time consuming due to the search of the matching between range and domain blocks. In order to improve this compression method, we propose firstly, a fast method for reducing the computational complexity of fractal encoding by reducing the size of the domain pool. This reduction is based on the lowest horizontal and vertical DCT coefficients of domain blocks. The experimental results on the test images show that the proposed method reduces the time computation and reaches a high speedup factor without decreasing the image quality. Secondly, we combine our method to the AP2D approach which uses two domain pools in two steps of encoding. A more reduction of encoding time is obtained without decreasing the image quality.

Key Words: Fractal image compression, Complexity reduction, DCT, SSIM index.

1 Introduction

Fractal image compression (FIC) was introduced by Barnsley and Jacquin [1-2] and it is one of the recent methods of compression. It has generated much interest due to its promise of high compression ratios and for being a simple and a very fast decompression method. 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 encode fractal images by means of some contractive transformations defining an Iterated Function System (IFS). As natural signals do not often possess global self transformability, Jacquin [2] proposed to look for local or partial transformability what led to the first algorithm of compression by Local or Partitioned Iterated Function Systems (PIFS).

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 different scale (domain block) looked for in the image. The classical encoding method, the full search, is time consuming because for every range block the corresponding block is looked for among all the domain blocks, i.e. the domain pool. Several methods are proposed to reduce the encoding time and the most common approach is the classification scheme [3-8]. 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 [3] constructed 72 classes for the image blocks according to their variances

Correspondence to: <sofia_douda@yahoo.fr>

Recommended for acceptance by <Chulhee Lee>
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

and intensities. In Wang et al. [8], four types of range blocks are defined based on the edge of the image. Jacobs et al. [9] uses skipping adjacent domain blocks whereas 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 the given range block [10]. Methods based on reduction of the domain pool are also developed. Saupe's lean domain pool method discards a fraction of domain blocks with the smallest variance [11] and Hassaballah et al.'s method removes the domain blocks with high entropies from the domain pool [12]. Other approaches produce improvements of FIC by tree structure search methods [13, 14], parallel search methods [15, 16] or by using two domain pools in two steps of FIC (AP2D) [17]. Also, the spatial correlation in both the domain pool and the range pool is added to improve FIC as developed by Truong et al. [18]. In these methods, high speedup factors are often associated with some loss of reconstructed image quality. In the present work, a new method to reduce the encoding time of FIC using the lowest horizontal and vertical DCT coefficients of domain blocks is proposed. This method speed up the encoding time by discarding the domain blocks having a low activity. The activity of the blocks is determined by the lowest horizontal and vertical DCT coefficients. The advantage of the proposed method is that it reaches a high speedup factor without decreasing the image quality. For more improvement, the proposed method is combined to the AP2D approach which uses two domain pools. In this combined method, the blocks having a low activity are discarded from the two domain pools.

2 Reduction of the encoding time based on DCT

2.1 DCT

By using DCT transformation, an image block can be transferred from the spatial domain to the frequency domain, in which the DCT coefficients located in the upper-left represent the low frequency information of the image block and reflect the rough contour of the image block. In contrast, the DCT coefficients located in the lower-right represent the high frequency information of the image block and reflect the fine texture of the image block.

Let D be a given image block of size N×N. The DCT of D, denoted by DCT_D, is computed from the formula [19]:

$$DCT_D(m,n)=\frac{2}{N}C_x C_y \sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1} D(i,j)\cos\left(\frac{(2i+1)m\pi}{2N}\right)x\cos\left(\frac{(2j+1)n\pi}{2N}\right) \tag{1}$$

Where m, n= 0, 1, …, N-1, and

$$C_m = \begin{cases} 1/\sqrt{2}, & \text{if } m=0 \\ 1, & \text{else} \end{cases}$$

The magnitude of $DCT_D(1,0)$ reflects the intensity variation between the left half and the right half of image block D. Similarly, the magnitude of $DCT_D(0,1)$ reflects the intensity variation between the upper half and the lower half of D. Then if both $|DCT_D(1,0)|$ and $|DCT_D(0,1)|$ are small, then the block D tends to have less edge structure. When a block has a high degree of edge structure, either $|DCT_D(1,0)|$ or $|DCT_D(0,1)|$ will be large. If $|DCT_D(1,0)|$ is larger, D will have horizontal edge properties. On the other hand, if $|DCT_D(0,1)|$ is larger, then D will have vertical edge properties. Finally, those blocks with high magnitudes of $|DCT_D(1,0)|$ and/or $|DCT_D(0,1)|$ are designed as high activity blocks and those with small magnitudes of $|DCT_D(1,0)|$ and $|DCT_D(0,1)|$ are designed as low activity blocks. The activity of each block D is determined as follows:

If $|DCT_D(1,0)| < T_{DCT}$ and $|DCT_D(0,1)| < T_{DCT}$
then D is a low activity block (2)
else D is a high activity block

Where |.| denotes the absolute value of its variable and T_{DCT} is a threshold.

Thus, an image block D can be determined as belonging to high activity or low activity type only by using its lowest vertical coefficient $DCT_D(1,0)$ and its lowest horizontal coefficient $DCT_D(0,1)$.

2.2 The proposed method

The most computationally intensive part of FIC process is the search step. The way to reduce the encoding time consists in decreasing the number of comparisons between each range block and the blocks in the domain pool. The proposed method reduces the encoding time by reducing the cardinal of the domain pool. The idea of this reduction scheme is based on the observations that only a fraction of the domain pool is used in the fractal encoding and that the set of used domain blocks is localized along edges and in the regions of high contrast of the image (designed as high activity blocks) as shown in figure 1. Consequently, there is a very large set of domain blocks with a low activity which is not used in the fractal code. The experimental results illustrated in figure 2 show that a large fraction of the domain pool has small lowest horizontal and vertical DCT coefficients while there is no tendency in the distribution of the used domain blocks. Therefore, it is possible to reduce the search time by discarding a large fraction of low activity blocks. In the proposed method, each range block is compared only with the domain blocks having a high activity. This method of reduction of the domain pool is simple since only few computations are required to calculate the lowest DCT coefficients $|DCT_D(1,0)|$ and $|DCT_D(0,1)|$ of a domain block D.

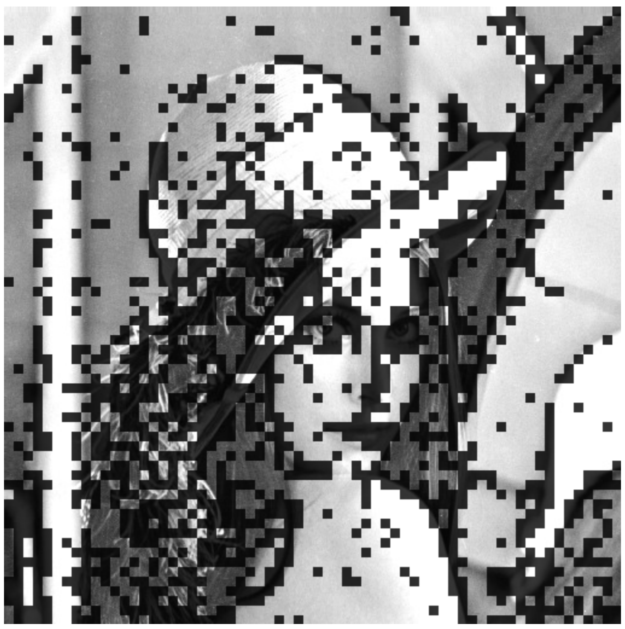
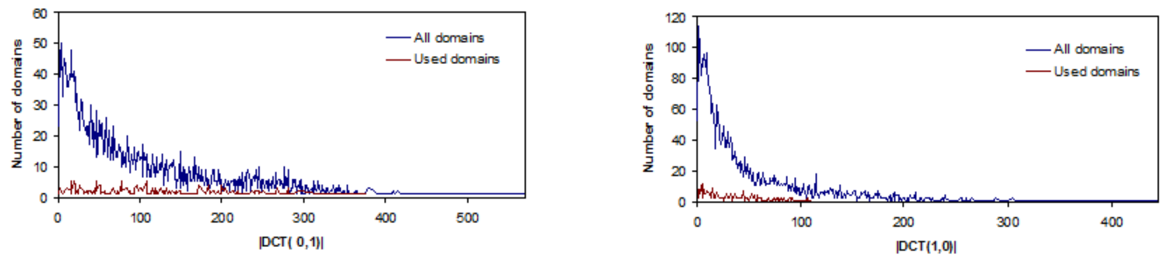
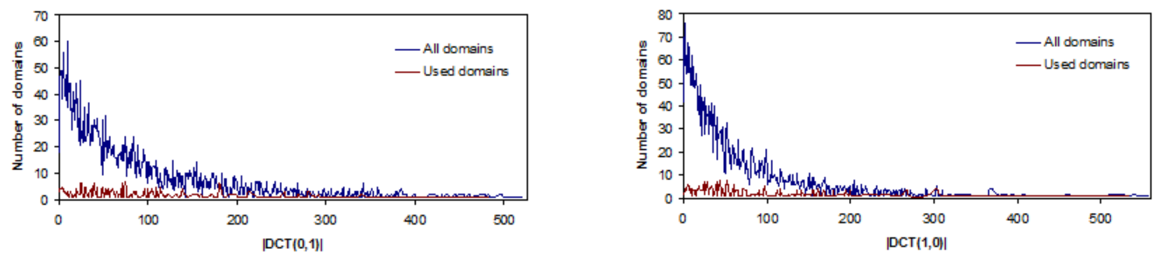


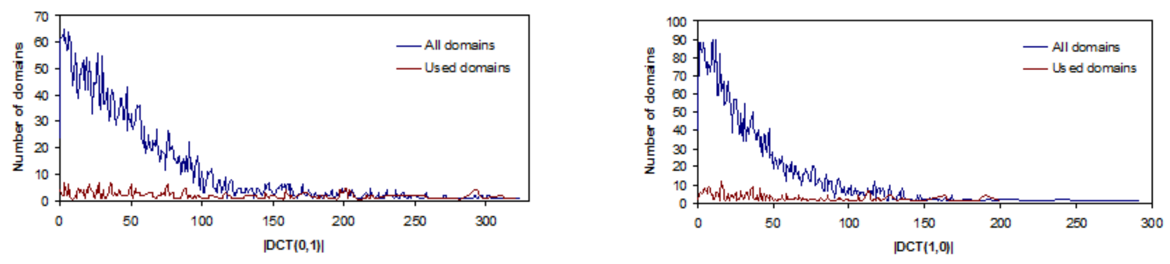
Fig. (1) Domain blocks of size 8x8 that are used for fractal encoding of Lena image are shown in black.



(a)



(b)



(c)

Fig. (2) Distribution of the lowest DCT coefficients |DCT(0,1)| and |DCT(1,0)| of all domain blocks of size 8x8 versus that of used domains in quadtree partitionning for Lena (a), Peppers (b) and Baboun (c).

2.3 Adaptive fractal encoding

The proposed method based on reduction of the cardinal of the domain pool given in section 2.2 is now applied to speed up the fractal image encoding. In the first stage of the proposed method, the domain pool is constructed and the domain blocks with a low activity are discarded.

The threshold T_{DCT} indicated in (2) can be fixed or chosen in an adaptive way. Determining T_{DCT} adaptively allow us to choose the speedup ratio. The main idea is to set the thresholds such that a fraction α of the domain pool can be discarded. Due to the fact that the encoding time depends on the number of comparisons between range and domain blocks, the speedup ratio can be estimated.

The determination of the threshold T_{DCT}, which depends on the fraction α of the domain pool to be eliminated, is summarised as follows:

1. For each domain block D, calculate the lowest DCT coefficients $DCT_D(1,0)$ and $DCT_D(0,1)$. Set $S = \max(|DCT_D(0,1)|, |DCT_D(1,0)|)$.
2. Sort all the values of S in increasing order.
3. Find S* corresponding to the value of α. Set the threshold $T_{DCT} = S^*$.

Due to the fact that we apply our method in the case of a quadtree partitioning, we choose different thresholds for every size of the domain blocks.

The first steps of the proposed method are as follows:

- Choose a value of α.
- Construct the domain pool D.
- Compute the two lowest DCT coefficients $DCT_D(1,0)$ and $DCT_D(0,1)$ for each domain block.
- Determine the threshold T_{DCT} for each size domain block.
- Remove the domain blocks which have a low activity from D.

3 Experimental results

The different tests are performed on three images, Lena, Peppers and Baboun (figure 3) with 8 bpp and the software simulation is done using C++ on a Windows XP, Intel Pentium Dual 2.16 Ghz platform. The quadtree partition [3] is adopted. The image quality is measured by the peak signal to noise ratio (PSNR) and the structural similarity Measure (SSIM) index [20].

The PSNR of the original image X and the distorted image Y of sizes N, is defined as follows:

$$PSNR = 10 \, x \log_{10}\left(\frac{255^2}{MSE}\right) \tag{3}$$

where

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 , \tag{4}$$

x_i and y_i are the gray levels of pixel of X and Y respectively.

The SSIM index is a method for measuring the similarity between two images X and Y defined by Wang [20] as follows:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{5}$$

where $\mu_X = \frac{1}{N}\Sigma x_i$, $\mu_Y = \frac{1}{N}\Sigma y_i$, $\sigma_X = (\frac{1}{N-1}\sum(x_i - \mu_X)^2)^{\frac{1}{2}}$, $\sigma_Y = (\frac{1}{N-1}\sum(y_i - \mu_Y)^2)^{\frac{1}{2}}$, $\sigma_{XY} = \frac{1}{N-1}\sum(x_i - \mu_X)(y_i - \mu_Y))$.

C_1 and C_2 are positive constants chosen to prevent unstable measurement when $(\mu_X^2 + \mu_Y^2)$ or $(\sigma_X^2 + \sigma_Y^2)$ is close to zero. They are defined in [20] as:

$$C_1 = (K_1L)^2 , C_2 = (K_2L)^2 \tag{6}$$

where L is the dynamic range of pixel values (L= 255 for 8-bit gray scale images). K_1 and K_2 are the same as in [20]: $K_1 = 0.01$ and $K_2 = 0.03$.

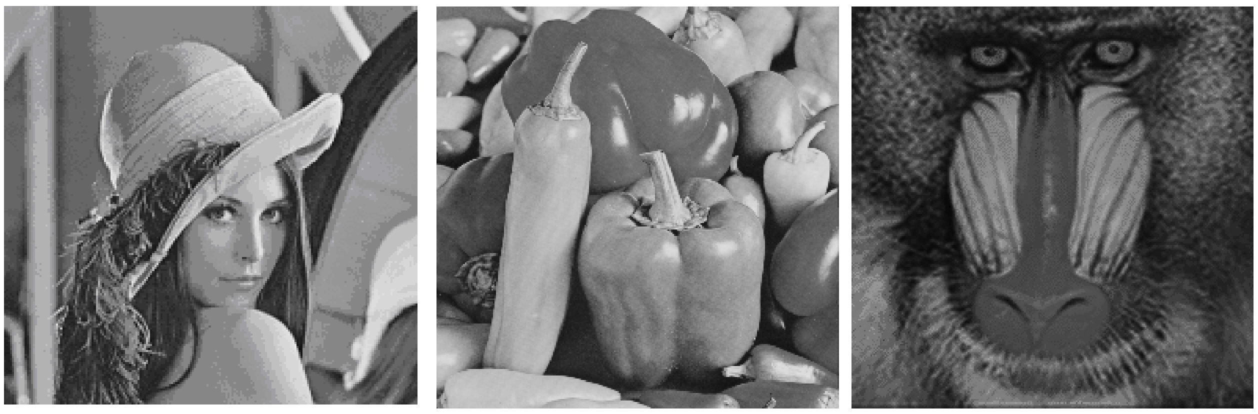
In the present work, we use an average SSIM (MSSIM) index to evaluate the overall image quality:

$$MSSIM(X,Y) = \frac{1}{M}\sum_{i=1}^{M} SSIM(x_i, y_i) \tag{7}$$

where X and Y are the original and the distorted images respectively; x_i and y_i are the image contents at the i^{th} local window and M is the number of local windows of the image.

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.,

$$SF = \frac{\text{Time taken in full search}}{\text{Time taken in a particular method}} \tag{8}$$



(a) (b) (c)

Fig. (3): Images of size 256x256 : Lena (a), Peppers (b) and Baboun (c).

Table 1 gives the encoding time, CR, PSNR and MSSIM measured on the three test images for different values of (1-α). The FS occurs when α=1 and there is no time reduction because no domain block is eliminated. As illustrated in figure 4, the encoding time scales linearly with α. When two thirds of the domain pool are discarded, the proposed method reach a SF larger than 3 with a slight increase of PSNR of 0.04 dB for Lena and 0.12 dB for Baboun and a drop of PSNR of 0.02 dB for Peppers. It also decreases the CR slightly for the three test images.

When $\alpha \geq 0.1$, there is a slight decrease of PSNR, CR and MSSIM while the SF reaches a value more than 8. For comparison, the FS reaches a PSNR of 30.92 dB and a MSSIM of 0.8909 with a required time of 18.85 seconds for Lena. In the proposed method, the encoding time of Lena is 2.24 seconds while PSNR is 30.41 dB and MSSIM is 0.8872 when α=0.1. The SF attains 8.42 with a drop of 0.52 dB, 0.0037 and 1.24 of PSNR, MSSIM and CR respectively. When $\alpha \leq 0.04$, a high time reduction is obtained but with a decrease of CR of 1.79, 2.07 and 1.15 for Lena, Peppers and Baboun respectively. This could be explained by the fact that some large range blocks could be covered well by some domain blocks which were excluded from the domain pool by our method. Therefore, these large range blocks are subdivided in four quadrants resulting in a decrease of CR. For a SF of 16, the quality of the test images is still preserved as shown in figure 5, 6 and 7.

Table 1. Encoding time, CR, PSNR and MSSIM when a fraction (1-α) of domain pool is discarded.

α	Lena					Peppers					Baboun				
	Time	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**
1	18.85	10.46	30.92	0.8909	1.00	18.50	10.98	31.91	0.8931	1.00	24.59	7.46	32.55	0.8802	1.00
0.9	16.99	10.41	30.94	0.8916	1.11	16.56	10.98	31.91	0.8931	1.12	22.00	7.45	32.56	0.8805	1.12
0.8	15.07	10.37	30.96	0.8924	1.25	14.91	10.94	31.92	0.8935	1.24	19.63	7.41	32.58	0.8812	1.25
0.7	13.18	10.30	31.00	0.8935	1.43	12.91	10.92	31.92	0.8938	1.43	17.16	7.39	32.58	0.8815	1.43
0.6	11.45	10.19	31.03	0.8941	1.65	11.10	10.85	31.93	0.8936	1.67	14.79	7.34	32.60	0.8824	1.66
0.5	9.61	10.11	31.04	0.8948	1.96	9.35	10.70	31.96	0.8954	1.98	12.44	7.30	32.63	0.9834	1.98
0.4	7.73	9.99	31.03	0.8950	2.44	7.59	10.51	32.01	0.8963	2.44	10.01	7.26	32.64	0.8828	2.46
0.3	6.03	9.71	30.96	0.8939	3.13	5.72	10.47	31.89	0.8946	3.23	7.53	7.21	32.67	0.8832	3.27
0.2	4.12	9.51	30.82	0.8929	4.58	3.94	10.28	31.76	0.8923	4.70	5.15	7.15	32.63	0.8820	4.77
0.1	2.24	9.22	30.41	0.8872	8.42	2.15	9.91	31.47	0.8877	8.60	2.74	6.98	32.53	0.8801	8.97
0.08	1.87	9.20	30.23	0.8837	10.08	1.80	9.83	31.27	0.8859	10.28	2.27	6.91	32.47	0.8785	10.83
0.06	1.51	9.10	30.16	0.8836	12.48	1.44	9.67	31.13	0.8842	12.85	1.80	6.80	32.43	0.8774	13.66
0.04	1.12	8.98	29.88	0.8807	16.83	1.08	9.48	30.90	0.8813	17.13	1.38	6.64	32.22	0.8737	17.82
0.02	0.74	8.80	29.66	0.8757	25.47	0.72	9.13	30.58	0.8764	25.69	0.91	6.49	31.93	0.8692	27.02
0.008	0.52	8.75	29.28	0.8683	36.25	0.54	9.02	29.99	0.8708	34.26	0.67	6.32	31.66	0.8628	36.70
0.006	0.51	8.67	29.17	0.8661	36.96	0.52	8.94	29.92	0.8692	35.58	0.64	6.31	31.25	0.8535	38.42

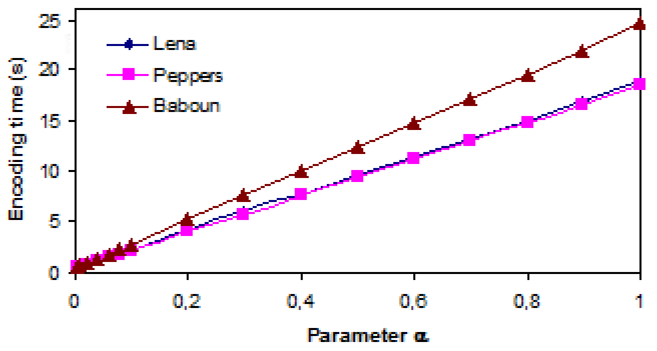
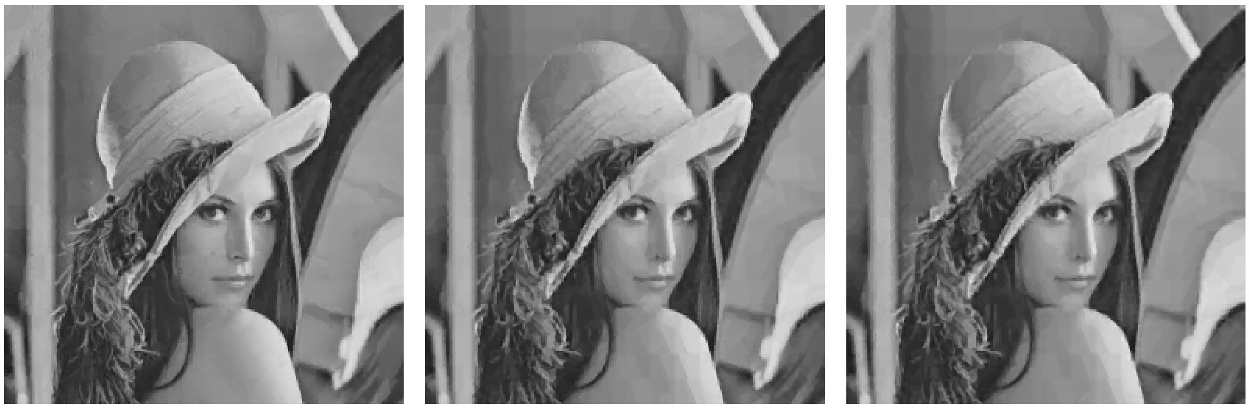
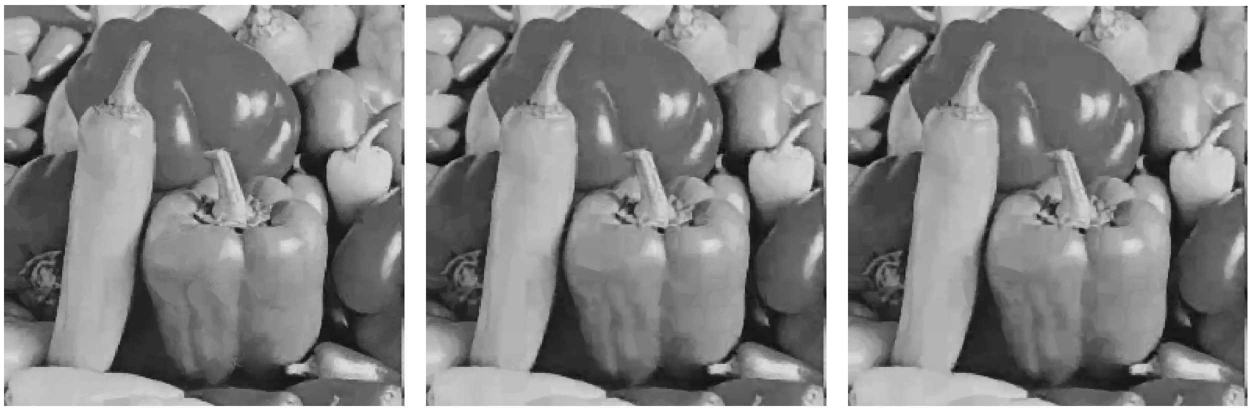


Fig. (4): Effect of parameter α on the encoding time for the test images.



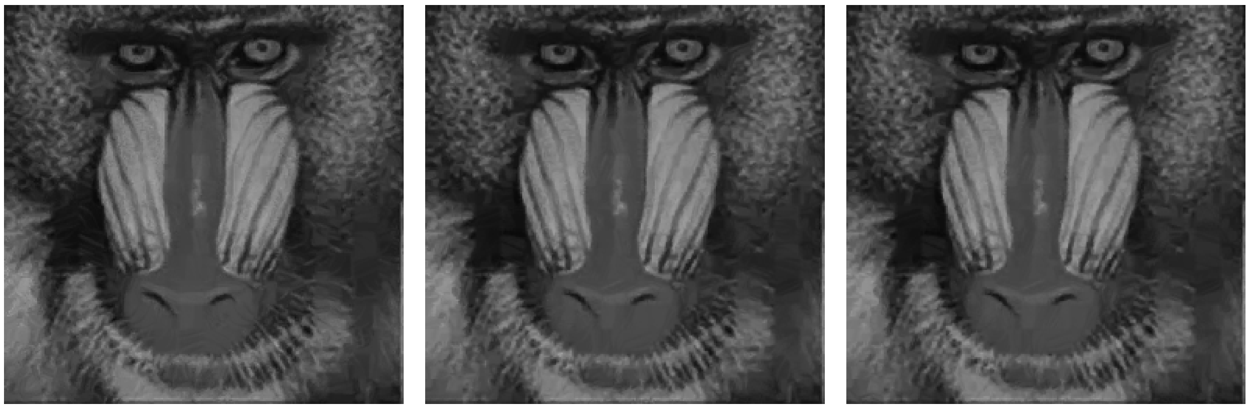
(a) SF = 1 (b) SF = 12.48 (c) SF = 16.83

Fig. (5): Reconstructed image Lena by full search (a): PSNR = 30.92 dB, MSSIM = 0.8909 and by the proposed method (b): PSNR = 30.16 dB, MSSIM = 0.8836 and (c): PSNR = 29.88 dB, MSSIM = 0.8807.



(a) SF = 1 (b) SF =12.85 (c) SF =17.13

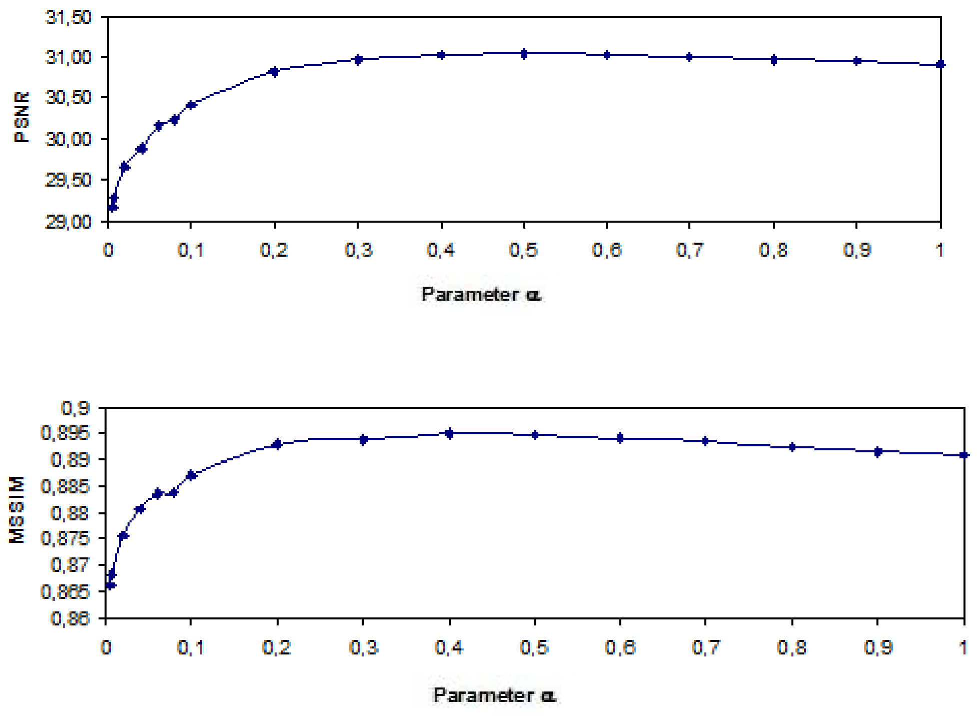
Fig. (6): Reconstructed image Peppers by full search (a): PSNR=31.91, MSSIM= 0.8931 and by the proposed method (b): PSNR= 31.13 dB, MSSIM = 0.8842 and (c): PSNR=30.90 dB, MSSIM=0.8813.



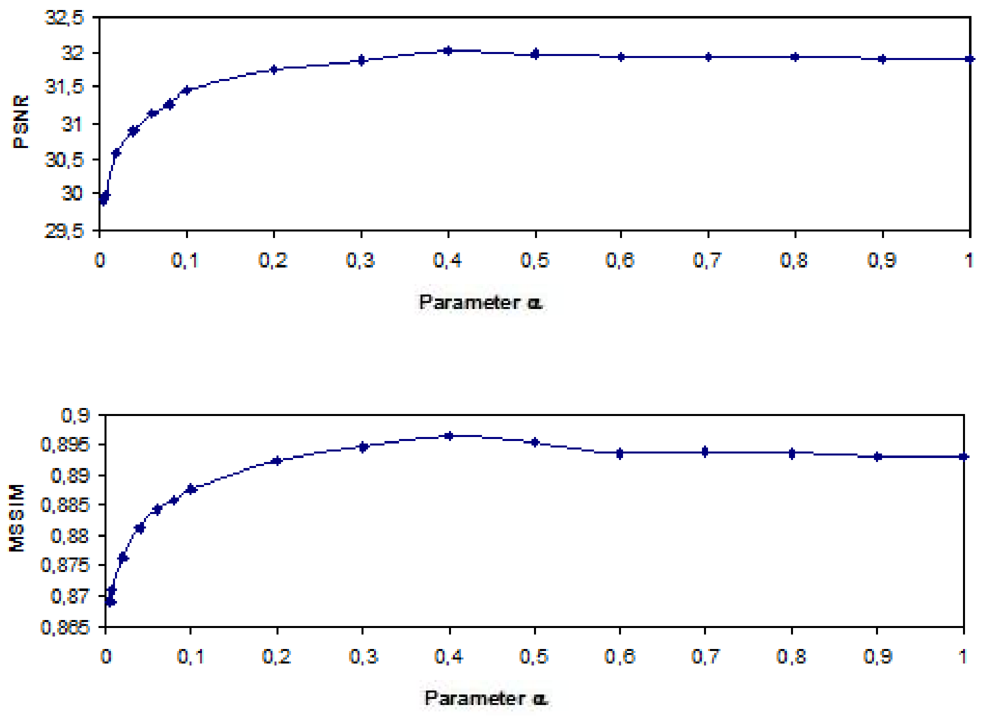
(a) SF = 1 (b) SF = 13.66 (c) SF = 17.82

Fig. (7): Reconstructed image Baboun by full search (a): PSNR = 32.55 dB, MSSIM = 0.8802 and by the proposed method (b): PSNR= 32.43 dB, MSSIM=0.8774 (c): PSNR= 32.22 dB, MSSIM=0.8737.

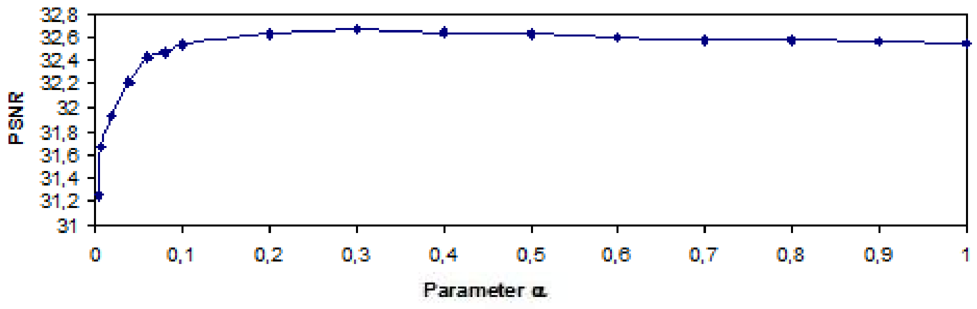
As shown in figure 8, PSNR and MSSIM vary in the same way according to the parameter α for the test images.

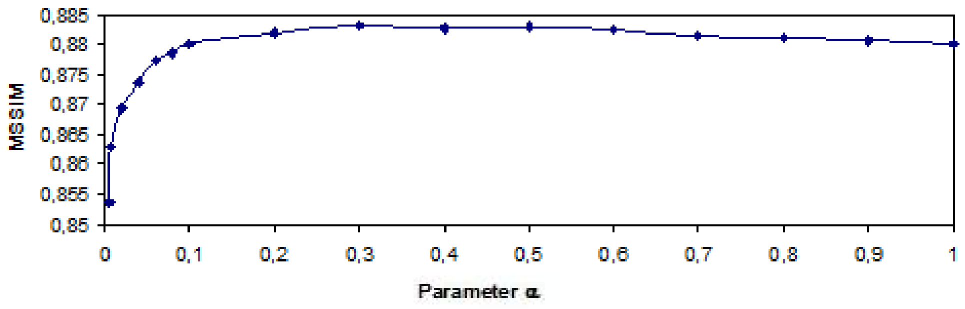


(a)



(b)





(c)

Fig. (8): Effect of parameter α on PSNR and MSSIM for Lena (a), Peppers (b) and Baboun (c).

To compare our approach with Hassaballah et al.' method (HM), we compute CR in addition to PSNR and MSSIM. We find that our method preserves well the image quality and marks a slight decrease of CR for a high SF than HM. For Lena, a SF of 10.08 is reached with a drop of PSNR of 0.69 dB and a decrease of CR of 1.26 while HM cause a drop of 3.69 dB of PSNR and a drop of CR of 3.28 for a SF of 9.74. For the same SF, the drop of MSSIM is of 0.0072 by our method and 0.0668 by HM. Furthermore, the results of encoding are still better than HM when the SF achieve a high value for the three test images.

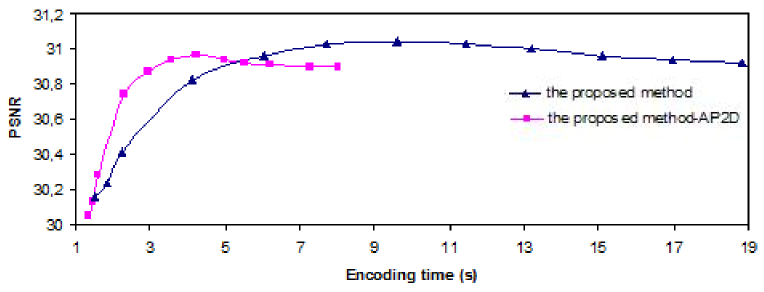
We also compare our method to Saupe's method (SM) and observe an improvement of the encoding time. For example, with Lena, a SF of 25.45 results in a PSNR of 29.66 dB, a CR of 8.80 and a MSSIM of 0.8757, while by SM a SF of 21.81 generate a PSNR of 29.30 dB, a CR of 8.51 and a MSSIM of 0.8654.

The comparison with AP2D-ENT [21] shows that our proposed method preserves the image quality and gives better results than those obtained by AP2D-ENT, precisely when SF is greater than 19. Indeed for Lena, the highest SF reaches 36 which generates a PSNR of 29.17 dB, a CR of 8.67 and a MSSIM of 0.8661. While the highest SF obtained by AP2D-ENT is 24 which generates a PSNR of 28.63 dB, a CR of 8.84 and a MSSIM of 0.8507. Similar improvements are observed for Peppers and Baboun.

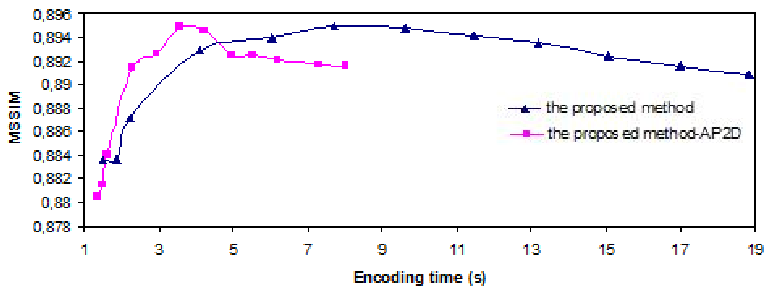
For more improvement, we combine the proposed method to AP2D. In this combination, we use two domain pools instead of one and encode an image in two steps as in [17]. The blocks having a low activity are eliminated from the two domain pools. Table 2 shows the results obtained by this combination. The main result consists in an improvement of SF precisely when $\alpha \geq 0.06$. For $\alpha < 0.06$, no change in the results is observed because the cardinal of the domain pool becomes too small. For $\alpha \geq 0.3$, SF increase twofold (fig. 9) with a maximum drop of PSNR of 0.09 dB for Lena, 0.9 dB for Peppers and 0.04 dB for Baboun. For MSSIM, a maximum drop of 0.0032 is observed for Peppers whereas a maximum increase of 0.0007 and 0.0016 is observed for Lena and Baboun respectively. Furthermore, this combined method preserves well the image quality as shown in figure 10.

Table 2. Encoding time, CR, PSNR and MSSIM for the proposed method combined to AP2D.

α	Lena					Peppers					Baboun				
	Time	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**
1	8.02	9.95	30.90	0.8916	2.35	7.25	10.36	31.85	0.8907	2.55	7.97	6.96	32.58	0.8818	3.09
0.9	7.29	9.95	30.90	0.8917	2.59	6.41	10.36	31.85	0.8908	2.89	7.28	6.96	32.59	0.8818	3.38
0.8	6.19	9.92	30.91	0.8921	3.05	5.91	10.33	31.84	0.8909	3.13	6.64	6.95	32.60	0.8819	3.70
0.7	5.52	9.86	30.92	0.8925	3.41	5.14	10.28	31.84	0.8911	3.60	5.84	6.94	32.61	0.8823	4.21
0.6	4.98	9.78	30.94	0.8925	3.79	4.51	10.18	31.89	0.8920	4.10	5.15	6.91	32.63	0.8828	4.77
0.5	4.23	9.68	30.97	0.8946	4.46	3.93	10.13	31.91	0.8929	4.71	4.42	6.88	32.66	0.8834	5.56
0.4	3.56	9.58	30.94	0.8949	5.29	3.30	10.01	31.92	0.8931	5.61	3.68	6.86	32.64	0.8825	6.68
0.3	2.95	9.36	30.87	0.8926	6.39	2.70	9.97	31.83	0.8917	6.85	2.94	6.79	32.63	0.8821	8.36
0.2	2.31	9.07	30.74	0.8915	8.16	2.09	9.75	31.68	0.8898	8.85	2.29	6.73	32.60	0.8819	10.74
0.1	1.61	8.92	30.28	0.8841	11.71	1.56	9.60	31.37	0.8855	11.86	1.62	6.62	32.43	0.8771	15.18
0.08	1.48	8.88	30.13	0.8815	12.74	1.42	9.60	31.17	0.8830	13.03	1.52	6.59	32.35	0.8750	16.18
0.06	1.36	8.84	30.05	0.8805	13.86	1.31	9.34	31.04	0.8819	14.12	1.42	6.57	32.30	0.8748	17.32

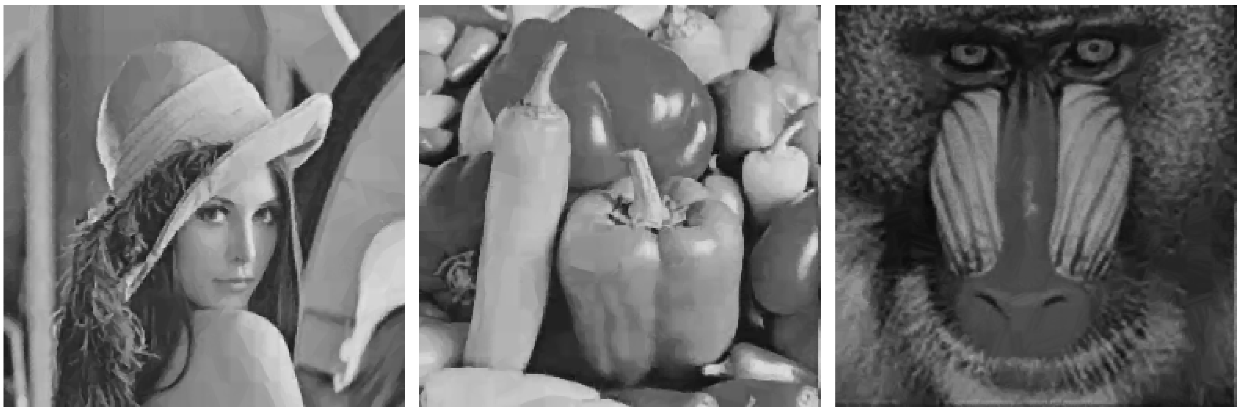


(a)



(b)

Fig. (9): Encoding time versus PSNR (a) and MSSIM (b) for Lena.



(a) PSNR=30.05 dB, MSSIM= 0.8805, SF = 13.86.

(b) PSNR=31.04 dB, MSSIM= 0.8819, SF = 14.12.

(c) PSNR=32.30 dB, MSSIM= 0.8748, SF = 17.32.

Fig. (10): Reconstructed image Lena (a), Peppers (b) and Baboun (c) when using AP2D and discarding the domain blocks having low activity.

In order to compare our combined method to Xing-yuan'method (X-YM) [22] which is based on spatial correlation and genetic algorithm, we apply our method in the case of a square partition with range block size 4x4 (Table 3). The comparison shows that our combined method reaches higher SFs than X-YM with less drops of PSNR. It generates the following SF values 109.49, 110.65 and 45.45 that are associated to drops of PSNR of 1.89, 2.15 and 1.16 for Lena, Peppers and Baboun respectively. Whereas by X-YM, the following SF values 72.68, 60.76 and 35.26 are associated to drops of PSNR of 2.05, 3.04 and 1.18 for Lena, Peppers and Baboun respectively.

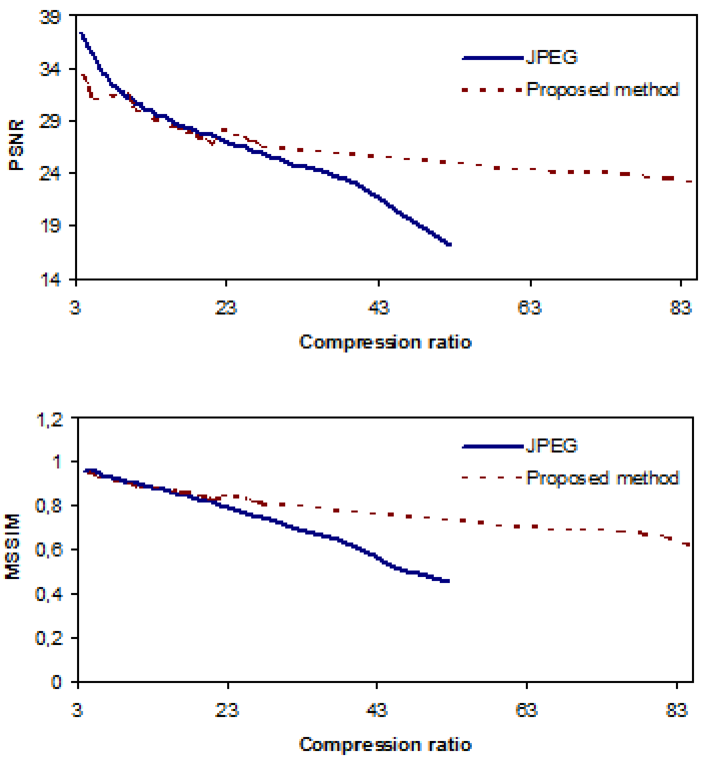
Table 3. The results of the combined method in the case of square partition with range block size 4x4 and when a fraction (1-α) of the two domain pools is discarded. The last line corresponds to the full search (FS).

α	Lena					Peppers					Baboun				
	Time	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**	**Time**	**CR**	**PSNR**	**MSSIM**	**SF**
1	30.73	4.41	32.72	0.9510	14.07	25.97	4.41	33.95	0.9578	16.02	22.38	4.41	34.86	0.9487	19.21
0.9	26.68	4.41	32.73	0.9511	16.21	23.26	4.41	33.94	0.9576	17.89	19.86	4.41	34.85	0.9486	21.65
0.8	23.42	4.41	32.73	0.9510	18.47	20.50	4.41	33.94	0.9575	20.30	18.07	4.41	34.84	0.9484	23.80
0.7	20.27	4.41	32.71	0.9506	21.34	18.16	4.41	33.90	0.9566	22.91	15.52	4.41	34.82	0.9480	27.71
0.6	17.18	4.41	32.70	0.9503	25.17	16.18	4.41	33.90	0.9566	25.71	13.32	4.41	34.79	0.9474	32.28
0.5	14.31	4.41	32.67	0.9496	30.22	12.83	4.41	33.87	0.9559	32.43	11.19	4.41	34.74	0.9465	38.43
0.4	11.90	4.41	32.61	0.9482	36.34	10.41	4.41	33.81	0.9545	39.97	9.46	4.41	34.67	0.9451	45.45
0.3	8.94	4.41	32.45	0.9456	48.38	8.20	4.42	33.65	0.9526	50.74	7.28	4.41	34.60	0.9438	59.06
0.2	6.49	4.43	32.18	0.9417	66.64	6.03	4.42	33.42	0.9496	69.00	5.30	4.42	34.52	0.9419	81.13
0.1	3.95	4.57	31.50	0.9316	109.49	3.76	4.51	32.85	0.9427	110.65	3.32	4.42	34.06	0.9345	129.52
0.08	3.48	4.63	31.27	0.9279	124.28	3.32	4.54	32.81	0.9412	125.32	3.00	4.42	33.96	0.9325	143.33
0.06	3.01	4.69	31.13	0.9259	143.68	2.89	4.61	32.35	0.9369	143.96	2.61	4.43	33.86	0.9303	164.75
0.04	2.50	4.75	30.89	0.9221	173.00	2.36	4.73	32.00	0.9322	176.29	2.27	4.44	33.39	0.9229	189.42
0.02	2.00	4.86	30.53	0.9165	216.25	1.91	4.87	31.49	0.9253	217.83	1.82	4.45	33.02	0.9156	236.26
0.008	1.65	5.05	29.98	0.9062	262.12	1.65	5.09	30.86	0.9157	252.15	1.64	4.50	32.57	0.9062	262.19
0.006	1.63	5.11	29.85	0.9035	265.33	1.59	5.20	30.60	0.9110	261.67	1.58	4.51	32.47	0.9047	272.15
FS	**432.49**	**4.13**	**33.39**	**0.9590**	**1.00**	**416.05**	**4.13**	**35.00**	**0.9958**	**1.00**	**429.99**	**4.13**	**35.83**	**0.9588**	**1.00**

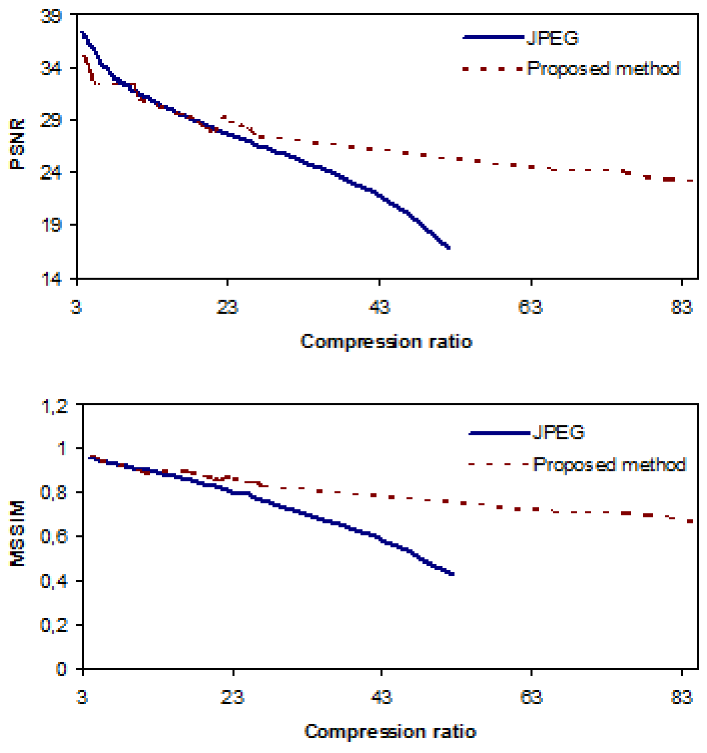
The proposed method is also compared with the conventional method JPEG [23]. In this comparison, the encoding time is not considered as a factor. PSNR and MSSIM versus compression ratio are presented in figures 11. The results show that:

- at a high CR, the proposed method is better than JPEG. With Lena for example, a CR of 65 generates a PSNR of 24.13 dB and a MSSIM of 0.6909 whereas by JPEG algorithm, a CR of 52.48 generates a PSNR of 17.13 dB and a MSSIM of 0.4513. Similar results are obtained for Peppers and Baboun.

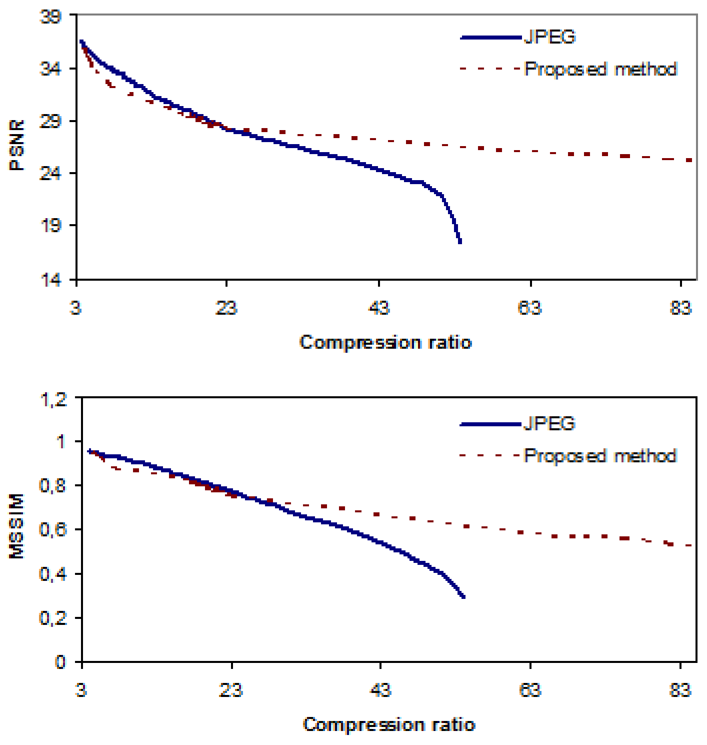
- at low CR, JPEG is better than the proposed method. The maximal drops of PSNR for Lena, Peppers and Baboun are 3.63 dB, 2 dB and 1.40 respectiveley. The maximal drops of MSSIM for Lena, Peppers and Baboun are 0.0118, 0.0072 and 0.046 respectively.



(a)



(b)



(c)

Fig. (11): Compression ratio versus PSNR and MSSIM for Lena (a), Peppers (b) and Baboun (c).

4 Conclusion

In this study, we propose firstly to reduce the time of fractal image encoding by using a new method based on the DCT coefficients. In this method, the domain blocks with a low activity are discarded from the domain pool. The activity of blocks is based on the lowest horizontal and vertical DCT coefficients. Experimental results show that the proposed method reaches a high speedup factor with a very little effect on the image quality and CR. Secondly, we propose to combine our proposed method to the AP2D approach to more improve the encoding time. The results obtained show an improvement of the speedup factor and no deterioration of the image quality.

References

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

[2] 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.

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

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

[5] 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.

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

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

[8] 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.

[9] 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.

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

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

[12] M. Hassaballah, M. M. Makky and Y. B. Mahdi. A fast fractal image compression method based entropy. Electronic Letters on Computer Vision and Image Analysis, Vol. 5, pp. 30-40, 2005.

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

[14] 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.

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

[16] 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.

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

[18] T. K. Truong, 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.

[19] N. Ahmed, T. Natarajan and K. R. Rao. Discrete Cosine Transform. IEEE Transactions on Computers, Vol. C-32, pp. 90-93, 1974.

[20] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, Vol. 13, No. 4, pp. 600–612, 2004.

[21] S. Douda, A. A. El Imrani and A. Bagri. A new approach for improvement of fractal image encoding. International Journal on Computer Science and Engineering, Vol. 02, No. 4, pp. 1387-1394, 2010.

[22] W. Xing-yuan, L. Fan-ping and W. Shu-guo. Fractal image compression based on spatial correlation and hybrid genetic algorithm. J. Vis. Commun. Image R. 20, pp. 505-510, 2009.

[23] Independent JPEG Group, http://www.ijg.org.