Speeding Up Fractal Image Compression by Reducing Image Size

By

Eman Abdul-Jabar Saad

Electronic Computer Center, Mustansiriyah University, Baghdad, Iraq

Abstract

In fractal compression the image to be encoded is partitioned into blocks called (ranges). Each range is coded by reference to some other part of the image called (domain) and by some affine transformation parameters. The number of ranges plays an important role in the compression ratio, encoding time and reconstructed image quality. In order to obtain high compression ratios, only a small number of blocks are allowed. Reducing the number of image partitions (Range Blocks) while keeping as much as possible the quality of the reconstructed image is the goal of this work (since, as the number of the ranges is reduced, the encoding time will be reduced and the compression ratio will be increased).In the proposed technique "Speeding up Fractal Image Compression (SFIC)", the image to be encoded is reduced to its quarter size then it will be partitioned to get about the quarter number of ranges produced in the traditional Fractal Image Compression (FIC), in the decoding stage the image will be decoded to its original size. From the experimental results we found that SFIC gives a high reduction in the encoding time and high increasing in the compression ratio with good reconstructed image quality.

Keywords: *fractal, partition, affine transforms,* Horizontal-Vertical Partitioning Technique (H-V PT).

الخلاصـة:

في ضغط الكسوريات، تقطع الصورة المراد تشفيرها الى قطع تسمى المديات. كل مدى (Range) يشفر بالاشارة الى جزء آخر من الصورة يسمى المجال المقابل (Domain) وبواسطة معاملات تحويلات تقاربية معينة. عدد المديات (Ranges) يلعب دور مهم في نسبة الضغط وسرعة التشفير وجودة الصورة المسترجعة.في التقنية المقدمة "تسريع ضغط الصور باستخدام الكسوريات بتصغير حجم الصورة"، يتم تصغير الصورة المراد تشفيرها الى ربع حجمها الاصلي وبذلك فإنها سوف تقطع الى حوالي ربع عدد المديات المتكونة في طريقة التقطيع التقليدية وسوف تعاد الصورة الى حجمها الاصلي في مرحلة فتح التشفير (الاسترجاع).من التجارب العملية تبين ان طريقة "تسريع ضغط الصور باستخدام الكسوريات" أعطت زمن تشفير قليل وزيادة كبيرة في نسبة الضغط مع جودة جيدة للصورة المسترجعة.

1. Introduction

One technique that has gained attention for its outstanding performance is fractal image compression FIC. This technique is based on the theory of IFS and its performance relies on the presence of self-similarity between the regions of an image. Since most images possess high degree of self-similarity, fractal compression contributes an excellent tool for compressing them [1]. Self-Similarity indicates that small portions of the image resemble larger portions of the same image. The search for this resemblance forms the basis of the fractal compression scheme [2]. Therefore image must be partitioned into blocks to find self-similar in other portion of the same image. FIC consists of finding a set of transformations that produces a fractal image which approximates the original image.The encoding process of fractal image compression is extremely computationally intensive (long encoding time). This weak aspect makes the fractal compression method still not widely used as standard compression. Although it has the advantage of fast decompression as well as very high compression ratios. These properties made it a very attractive method for applications in multimedia: for example, Microsoft adopted it for compressing thousands of images in its Encarta multimedia encyclopedia [3].In order to reduce the long encoding time of FIC while preserving the good quality of reconstructed image, which will lead to consider the FIC as one of the best compression method in terms of its compression ratio, encoding time, and image quality, we propose a new method that depends on the idea of reducing mapping search operation by suggesting a new searching mechanism. Speeding up Fractal Image Compression (SFIC) will reduce the number of image ranges to about quarter the number of the ranges produced using traditional FIC.

2. The proposed method

SFIC method can be expressed by the following stages:

2.1 Generate the Reduced Image and the range blocks

The original image of size (W x H) will be reduced to its quarter size to be an image of size (W/2 x H/2) using the average 2x2 down sampling method [4], as illustrated in figure (1). this process will be responsible for reducing the number of the image range blocks to about its quarter number using any partitioning techniques.

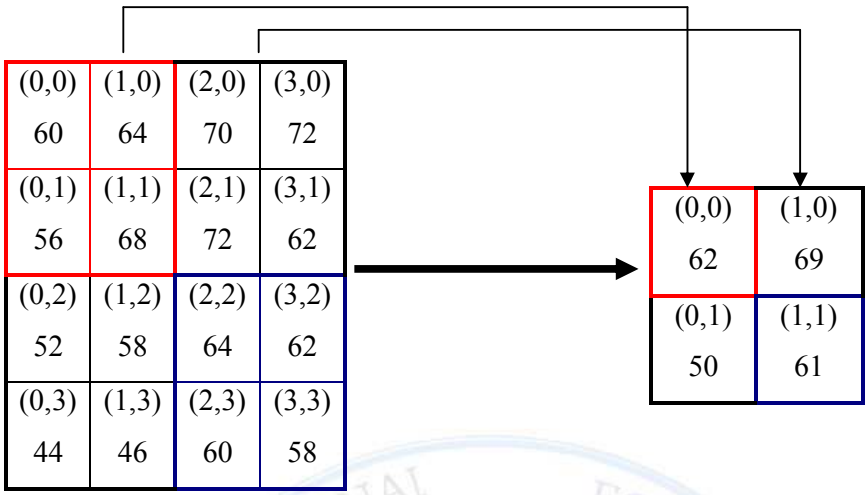


Fig.(1): shown down sampling method

Assume that the original image of size (256 x 256) is to be partitioned in FIC using fixed partitioning technique with range block size = 4, so the image will be partitioned into 4096 (256/4 x 256/4) range blocks of size 4, while the reduced image in SFIC which is of size (128 x 128) will be partitioned into 1024 (128/4 x 128/4) range blocks of size 4. This significant reduction in the number of range blocks of the image to be coded in SFIC will give higher compression ratio. On the other hand the encoding time also will be reduced significantly, because the computations needed in the encoding process will be also reduced to about its quarter number since the number of these computations is related to the number of the range blocks (for each range block R_i, the encoding process searches all the domain pool to find the domain block D_j that is best cover R_i).

2.2 Generate the Domain Pool

After generating the range pool from the reduced image, the domain pool will be created by down sampling the reduced image to its quarter size ($1/8^{th}$ the size of the original image) using the same method that illustrated in section (2.1). Figure (2) demonstrates this operation:

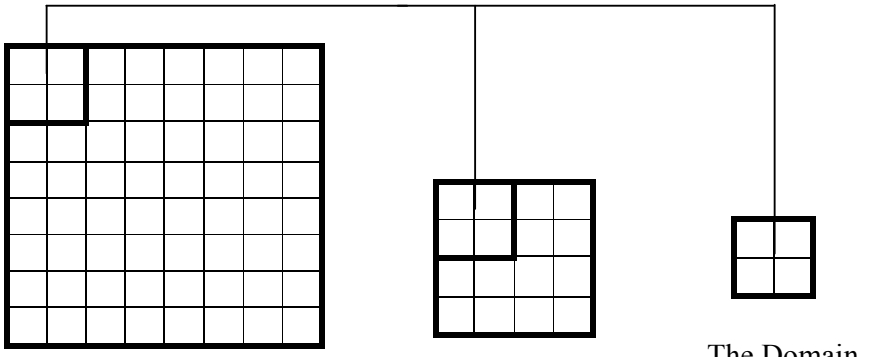


Fig.(2): Generate the reduced image and its domain

The reduction in the domain size to 1/8th the size of the original image will reduce the number of domain blocks D (Domain Pool) to about its quarter number as if it is created as the quarter size of the original image in the traditional FIC. To explain this fact consider the original image of size 256 pixels width x 256 pixels height is partitioned into 4096 (i.e. $R_1, R_2, \ldots, R_{4096}$) range blocks of size 4x4 pixels using fixed partitioning technique and if the encoder uses a domain step of size 4 to step through the domain array vertically and horizontally during the search process, then depending on the following equation:

$$D = \left(\frac{W_d - R_s}{S_{hv}} + 1\right) \times \left(\frac{H_d - R_s}{S_{hv}} + 1\right) \quad \ldots(1)$$

Where

W_d: The width of domain array.

H_d: The height of domain array.

R_s: The range size.

S_{hv}: The horizontal and vertical domain step through the domain.

The number of domain blocks D in the domain pool is:

1. If the domain array is created as quarter the size of the original image as in the traditional FIC (128 pixels width x 128 pixels height). Then the number of domain blocks will be:

$$D = \left(\frac{128-4}{4}+1\right)\times\left(\frac{128-4}{4}+1\right)=1024$$

2. If the domain array is created as $1/8^{th}$ the size of the original image (64 pixels width x 64 pixels height). Then the number of domain blocks will be:

$$D = \left(\frac{64-4}{4}+1\right)\times\left(\frac{64-4}{4}+1\right)=256$$

Coupling the significant reduction in the number of range blocks R and domain blocks D obtained by the suggested technique will give a significant reduction in the encoding time and a significant reduction in the size of the compressed file.

The consuming time of the encoding process occurs during the matching process since for each range block R_i in the range pool, all the domain blocks D in the domain pool (with all its 8 isometric symmetry) need to be searched to find $D_i \in D$ that best cover R_i.

In our example when the number of range blocks =4096 and the domain blocks =1024 (as in traditional FIC) then the encoder will need to 4096 x 1024 x 8 = 33,554,432 of matching operations to cover all range blocks, while in our suggested technique (SFIC) the encoder will need to 1024 x 256 x 8 = 2,097,152 of matching operations to cover them. This clear reduction in the matching operations is responsible for the high-speed encoding process achieved by this technique as the experimental results will show in the next section.

Also in addition to the reduction occurs in the number of bits required to encode the original image that is obtained from the reduction of the range blocks, a reduction in the number of bits needed to store each of x and y coordinates of the best matched domains will occur too, that is because of a smaller domain size will be used in our suggested technique. In our example when the domain of size (128 x 128), the maximum value for each x and y coordinates will be (124), by dividing it on the domain step (124/4 =31) then the encoder will need 5 bits to store each of x and y coordinates of the best matched domains, while in the domain of size (64 x 64) the maximum value for each x and y coordinates will be (60) by dividing it on the domain step (60/4 =15) then the encoder will need 4 bits to store each of x and y coordinates. And this will increase the compression ratio achieved by the reduction of

range blocks. So in our example the total number of bits required to save the IFS parameters for each range will be 25 bits (5 for x, 5 for y, 5 for s_q, 7 for o_q and 3 for s_{ym}) in traditional FIC while only 23 bits (4 for x, 4 for y, 5 for s_q, 7 for o_q and 3 for s_{ym}) required in SFIC.

Then the total number of bits required to store the IFS code of the original image in traditional FIC will be: 4096 x 25= 102,400 bits (12,800 bytes), while the total number of bits required to store the IFS code of the original image in SFIC will be: 1024 x 23= 23,552 bits (2,499 bytes). This means that the C.R. obtained by FIC = (65,536/ 12,800) = 5.12 while C.R. obtained by FRIC = (65,536/2,499) = 26.22.

2.3 Matching Process

After generating the range and domain pools, the matching process implies for each range block, with all domain blocks (D) are listed in domain pool, by performing a set of affine transformations.

The best matching between domain and range blocks which are satisfied the minimum distortion error $E(R,D)$ (5):

$$\mathrm{E(R,D)} = \frac{1}{\mathrm{n}}\sum_{i=1}^{n}(s.d_i + o - r_i)^2 \qquad \ldots(2)$$

This will give us contrast and brightness setting that makes the affinely transformed D_i values has the least squared distance from R_i values. The minimum of $E(R,D)$ occurs when the partial derivatives with respect to the scale "s" and offset "o" parameters are zero, which occurs when (6):

$$\mathrm{S} = \frac{\left[n\left(\sum_{i=1}^{n} d_i r_i\right) - \left(\sum_{i=1}^{n} d_i\right)\left(\sum_{i=1}^{n} r_i\right)\right]}{\left[n\sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2\right]} \qquad \ldots(3)$$

And

$$\mathrm{O} = \frac{\left[\sum_{i=1}^{n} r_i \sum_{i=1}^{n} d_i^2 - \sum_{i=1}^{n} d_i \sum_{i=1}^{n} d_i r_i\right]}{\left[n\sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2\right]} \qquad \ldots(4)$$

In this case:

$$E(R,D)=\frac{1}{n}\left[\sum_{i=1}^{n} r_i^2+S\left(S\sum_{i=1}^{n} d_i^2-2\sum_{i=1}^{n} d_i r_i+2O\sum_{i=1}^{n} d_i\right)+O\left(nO-2\sum_{i=1}^{n} r_i\right)\right] \quad \ldots(5)$$

Where:

d_i : is the pixels values of the domain block.

r_i : is the pixels values of the range block.

n : is the number of pixels in each block (i.e. the block size).

The root mean square error is equal to $\sqrt{E(R,D)}$ (6).

Scale parameter s has limit interval value ranging [-2,2] and offset parameter o has limit interval value ranging [-255,255].

The match process implies that for all domain blocks D_i, listed in the domain pool (Ω), and computing the optimal approximation of error equation (5) with the eighth cases of symmetry operations (the symmetry index illustrated in table (1)).

Table (1): Symmetry index and corresponding type

Symmetry index	Type
0	Identity
1	Rotation by 90°
2	Rotation by 180°
3	Rotation by 270°
4	Reflection
5	Reflection and rotation by (-90°)
6	Reflection and rotation by (-180°)
7	Reflection and rotation by (-270°)

The optimal approximation for each range block could be done as follows:

1. Compute the scale (s) and offset (o) coefficients, using equations (3) and (4).
2. **Quantize** s and o coefficients:

   Since the IFS coefficients (i.e., s and o) are real values, and in order to increase the compression, they must be quantized before storage (Quantization is simply a

process of reducing the number of bits needed to store coefficients values by reducing its precision from float type to integer). So After assigning the number of bits allocated for *s* and *o* coefficients, Quantization is performed by using the uniform quantize equations:

$$s_q = round\left(\frac{(2^{bs}-1)}{\text{MaxScl}-\text{MinScl}}(s-\text{MinScl})\right) \quad \ldots(6)$$

$$o_q = round\left(\frac{(2^{bo}-1)}{\text{MaxOfs}-\text{MinOfs}}(o-\text{MinOfs})\right) \quad \ldots(7)$$

Where:

b_s: number of bits assigned to *s* coefficient.

b_o: number of bits assigned to *o* coefficient.

3. Check the quantized coefficients
   - Scaling coefficient:

     If $s_q <$ MinScl then $s_q =$ MinScl else if $s_q >$ MaxScl then $s_q =$ MaxScl
   - Offset coefficient

     If $o_q <$ MinOfs then $o_q =$ MinOfs else if $o_q >$ MaxOfs then $o_q =$ MaxOfs
4. Compute the approximation error *E(R,D)* using equation (5).
5. Compare the computed error with the minimum registered error (E_{min}): if $E(R,D) > E_{min}$ then jump to step 7 .
6. Register:

   s_q, o_q, s_{ym}=symmetry index, x_d and y_d (the coordinates of the current tested domain block).
7. Repeat the step (1) to (5) for all symmetry versions of the tested domain block.
8. Repeat step (1) to (7) for the next domain blocks in the domain pool.
9. To increase the compression ratio will be minimize the values x_d and y_d by using:

$$x = \frac{x_d}{s_{hv}} \quad \ldots(8)$$

$$y = \frac{y_d}{s_{hv}} \quad \ldots(9)$$

The output is the set of parameters (s_q, o_q, x, y and s_{ym}) as a set of fractal coding parameters (IFS - Code) for the tested range block.

In other words, fractal image encoding process implies the determination of all mapping parameters, and they are stored sequentially as array of set of fractal parameters. The length of this array is equal to number of range blocks in the range pool. The IFS mapping parameters are listed in table (2).

Table (2): Description of IFS code

Parameters	Description
x	x - Coordinate of the optimal matched domain blocks divided by S_{hv}.
y	y - Coordinate of the optimal matched domain blocks divided by S_{hv}.
s_q	Quantized value of scaling coefficient of the optimal matched domain block
o_q	Quantized value of Offset coefficient of the optimal matched domain block
s_{ym}	The transformation state of the best matched domain block

2.4 Decoding Process

Decoding process of SFIC technique will start with any initial image with twice the size of domain array produced in the encoding process. In our example since the domain array in the encoding process is of size (64 x 64), the decoding process will start with any initial image of size (128 x 128). By implementing the IFS parameters to the new corresponding domains to reconstruct ranges with twice the size of the corresponding ranges in the reduced image, an image of the size equal to the size of original image can be obtained. Then an average filter of (2 x 2) can be implemented on the resulted image to reduce the side effects of down sampling processes occur on the image during the encoding processes. Then the resulted image will be down sampling to produce the new domain and by iterating the all above process several times (8 iterations) the attractor image can be obtained.

The decoding process can be explained in the following steps:

1. Initializing the first domain pool with blank image or any available image but with twice the size of the encoding domain array.
2. Reading IFS codes (x, y, s_q, o_q, s_{ym}) from its file and:

- The values of x_d and y_d, must be determined by using:

$$x_d = 2 \times (x \times S_{hv}) \quad \ldots(10)$$

$$y_d = 2 \times (y \times S_{hv}) \quad \ldots(11)$$

- The values of the indices of (s_q) and (o_q) for each range block should be dequantized by using the dequantization equations (12) and (13).

$$s = \frac{(\text{MaxScl} - \text{MinScl})}{(2^{bs} - 1)} \text{s}_q + \text{MinScl} \quad \ldots(12)$$

$$o = \frac{(\text{MaxOfs} - \text{MinOfs})}{(2^{bo} - 1)} \text{o}_q + \text{MinOfs} \quad \ldots(13)$$

3. For each range R_i:

- Determining the position and the size of range block in the reduced image and then determining the new position and size of it in the reconstructed image to be as the same size of the original image. The new position and size can be computed as follows:

$$x_{nr} = 2 \times x_r$$

$$y_{nr} = 2 \times y_r$$

$$siz_{nr} = 2 \times siz_r$$

Where

x_r, y_r, and siz_r: Are the x and y coordinates and the size of the range in the reduced image respectively.

x_{nr}, y_{nr}, and siz_{nr}: Are the x and y coordinates and the size of the range in the reconstructed image respectively.

- Determining the position of the matched domain block D_i in the domain pool, then D_i block is transformed (rotated or reflected or both) according to its corresponding symmetry coefficient value.
- An optimal approximation of range done by decode the IFS data by multiplying the corresponding best matched domain block (D_i) by the scale value and adding to the result of the offset value, i.e. equation (14).

$$R = sD + o \quad \ldots(14)$$

4. Implementing an average filter of (2 x 2) on the reconstructed image.
5. Down sampling the reconstructed image to generate a new domain pool using averaging method.
6. Repeating steps (3-5) until the attractor is reached.

Figures (3a and 3b) describe the SFIC technique. And by iterating the process described in figure (3b) several times (8 iterations) an attractor of the original image can be obtained.

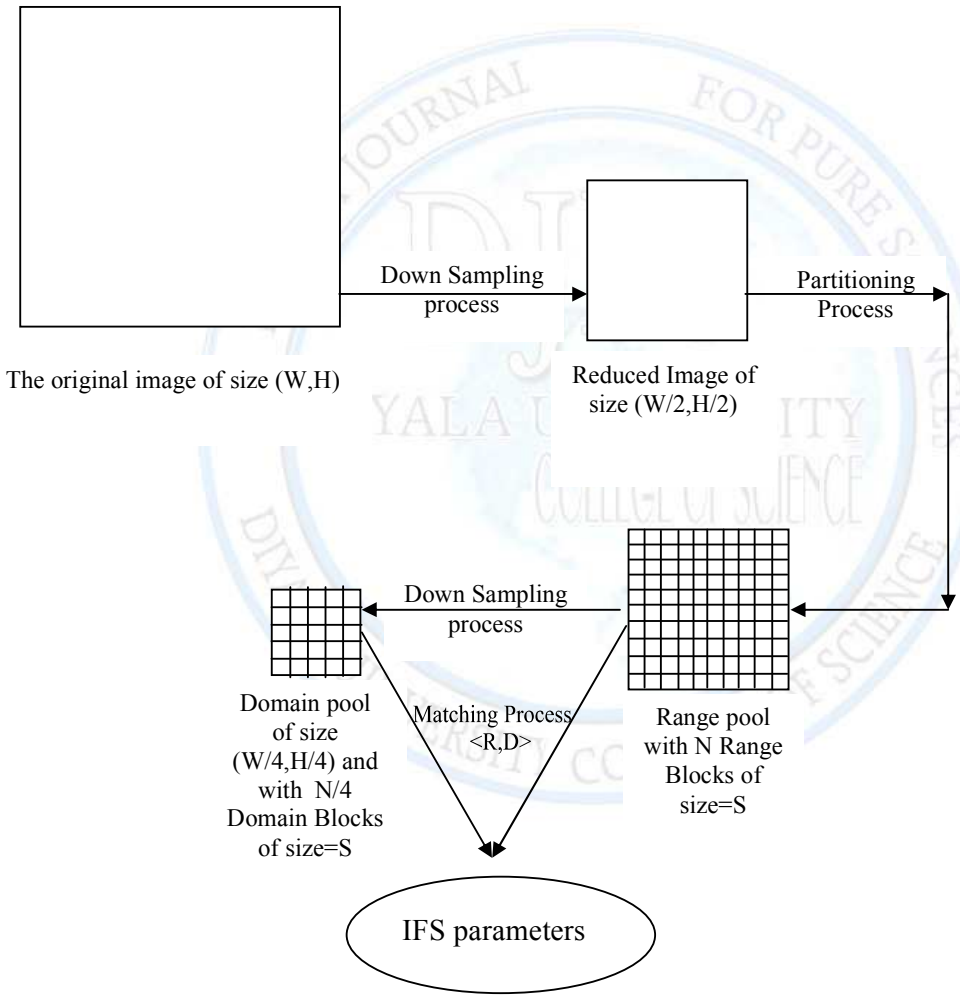


Fig.(3a): The encoding process of the SFIC Technique

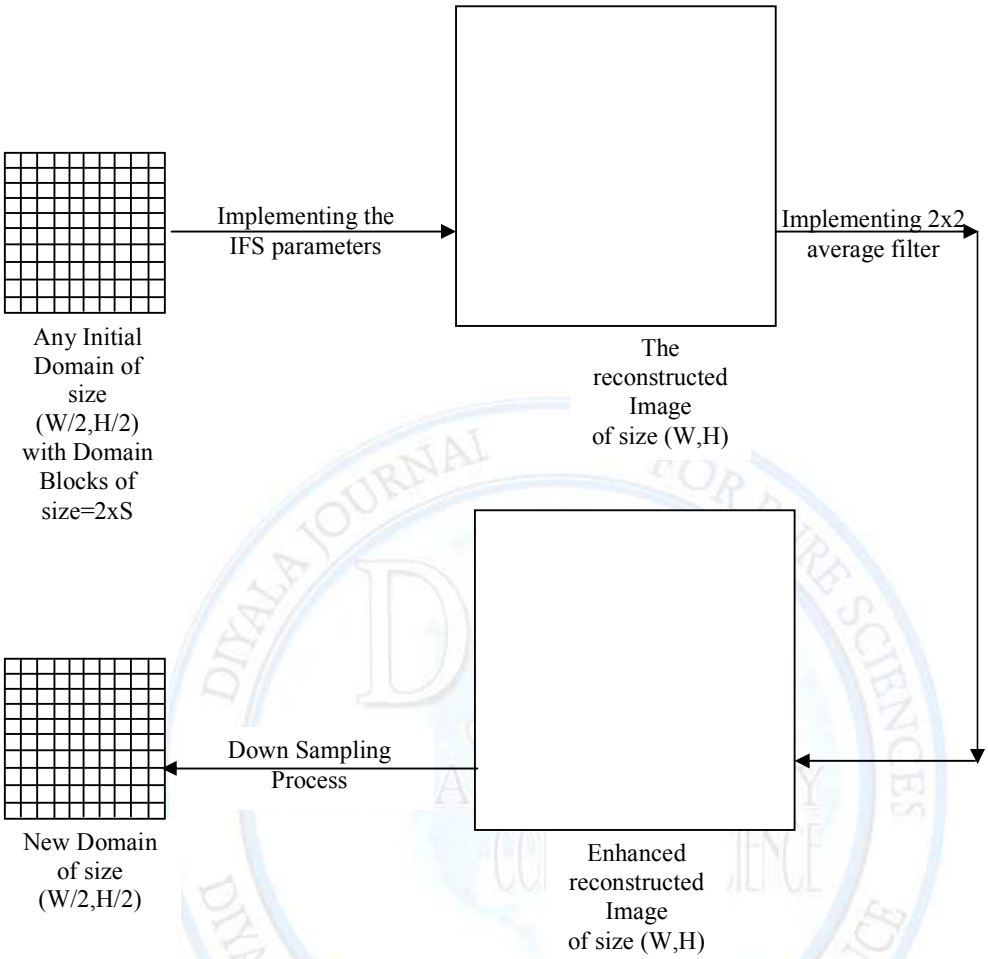


Fig.(3b): The decoding process of the SFIC technique

3. Experimental Results

The tests have shown that applying this technique leads to high speed encoding process (reducing the encoding time many times) and to get higher compression ratio, but the reconstructed image quality will be reduced. Figure (4) illustrates the comparison between FIC based on fixed partitioning with block size=4 and SFIC based on fixed partitioning technique with block size=4. From the figure it is obvious how the different in results is, the number of blocks in SFIC = 1/4 the number of blocks in FIC, the compression ratio in SFIC = 4^{th} times the compression ratio in FIC and SFIC is faster (about 94%) than FIC.

In hierarchal partitioning techniques we can make a type of balance among the compression ratio, encoding time and the reconstructed image quality that can be achieved using SFIC. From the tests we have found that the best value of the minimum block size=2 and the maximum block size=8 or 16. Figures (5 and 6) show the comparison between results obtained by applying FIC based on H-V partitioning technique and SFIC based on H-V partitioning technique on different test images.

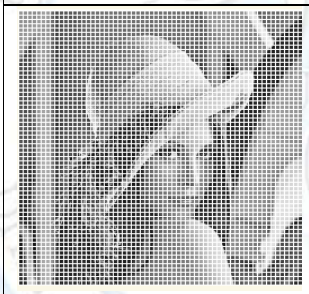
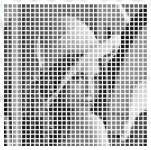
FIC	SFIC
Block size= 4x4	Block size= 4x4
	
No.of blocks=4096	No.of blocks=1024
Reconstructed Image	Reconstructed Image
	
C.R.=5.12 PSNR=31.61 E.T.=29.29 Sec.	C.R.=22.25 PSNR=24.92 E.T.=1.94 Sec

Fig.(4): Comparison between the results of FIC based on fixed partitioning and SFIC based on fixed partitioning applied on Lenna 256x256 image.

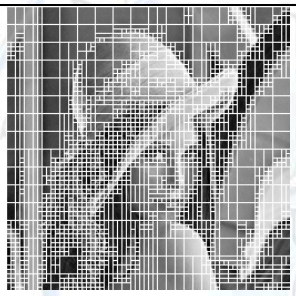
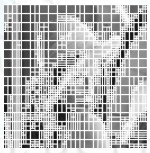
FIC	SFIC
MaxSiz=16 MinSiz=4 I_f=0.3 R_a=0.065	MaxSiz=8 MinSiz=2 I_f=0.6 R_a=0.0
	
No.of blocks=1947	No.of blocks=1430
Reconstructed Image	Reconstructed Image
	
C.R.=9.63 PSNR=30.97 E.T.=19 Sec.	C.R.=14.17 PSNR=26.26 E.T.=2.22 Sec.

Fig.(5): Comparison between the results of FIC based on H-V partitioning and SFIC based on H-V partitioning applied on 256x256 Lenna image.

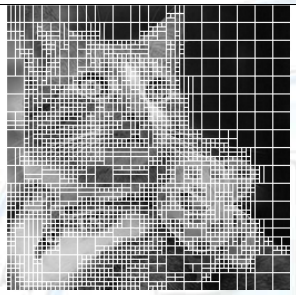
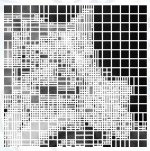
FIC	SFIC
MaxSiz=16 MinSiz=4 I_f=0.4 R_a=0.05	MaxSiz=8 MinSiz=2 I_f=0.4 R_a=0.03
	
No.of blocks=2075	No.of blocks=1691
Reconstructed Image	Reconstructed Image
	
C.R.=9.05 PSNR=31.55 E.T.=19.80 Sec.	C.R.=11.96 PSNR=28.05 E.T.=2.44 Sec.

Fig.(6): Comparison between the results of FIC based on H-V partitioning and SFIC based on H-V partitioning applied on 256x256 Cat image.

Table (3) shows the comparison in encoding results achieved by applying FIC based on H-V PT and the encoding results achieved by applying SFIC based on H-VPT on the test images. They show that applying SFIC based on H-VPT to the test images give about (24.17%)

reduction in the average No. of blocks and about (88.27%) reduction in the average encoding time and about (40.43%) increasing in the average compression ratio over applying FIC based on H-VPT but with some reduction in the PSNR about (13.25%). Also it is necessary to mentioned here that this percentage values will be changed by changing the test images.

Table (3): The averages of blocks No., C.R., PSNR and E.T. resulted from applying the FIC based on H-VPT and SFIC based on H-VPT on the previous test images.

	Image	FIC based on H-VPT	SFIC based on H-VPT
Number of Blocks	Lenna	1947	1430
	Cat	2075	1691
Average number of blocks		**1866**	**1415**
C.R.	Lenna	9.62	14.17
	Cat	9.05	11.96
Average C.R.		**10.51**	**14.76**
PSNR	Lenna	30.97	26.26
	Cat	31.55	28.05
Average PSNR		**33.57**	**29.12**
E.T.	Lenna	19.10	2.22
	Cat	19.85	2.44
Average E.T.		**18.67**	**2.19**

References

[1]: Mahadevaswamy H.R., "New Approaches to Image Compression", Ph.D. thesis, Regional Engineering College, university of Calicut, December, 2000.

[2]: Razaak H. H., "Adaptive Fractal Image Compression", M.Sc. thesis, College of Science, AL-Mustansiriya University, 2000.

[3]: Eman A. S., "Suggested Algorithm for Fractal Image Compression", M.Sc. thesis, College of Science, AL-Mustansiriya University, 200٥.

[4]: Lee, S.," Parallel Processing Architecture for Fractal Image Compression", Ph.D. thesis College of Engineering ", Tohoku University, Japan, 2000.

[5]: Hussein A. A., "Fractal Image Compression with Fasting Approaches", M.Sc. thesis, College of Science, Saddam University, 2003.

[6]: Fisher, Y., and Menlove, S., "Fractal Image Encoding with HV partitioning, Springier-Verlage, New York, 1994.