International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 7, Issue 1 (May 2013), PP. 52-61

Fractal Image Compression Based on Jointly and Different Partitioning Scheme

Jamila Harbi S [1], Salam A. N[2]
[1] Al-Mustenseriyh University, College of Sci., Computer Sci. Dept., Iraq.

Abstract:- Many researchers using HV-partitioning image in fractal encoding, because is the best way for reducing the encoding time with best quality of reconstructed image. In our paper we are proposed a new idea to partition the RGB-image after applied HV-partitioning depended onto two ways. First way using *Jointly Partitioning Scheme* is partitioned RGB image (i.e. all three channels Red, Green, and Blue) at one time depending on the information regions of image by using threshold value of partitioning. In this scheme, three channels have the same number of range blocks and the same range coordinates. The second using *Different Partitioning Scheme*. In this scheme, each channel will be partitioned depending on its information, and then we have different range blocks for each Red, Green, and Blue channel. The second idea is more effectively in encoding stage, because each channel has the different number of ranges.

Keywords:- HV partitioning, Fractal, Encoding process.

I. INTRODUCTION

Fractal image compression is a lossy data compression technique and exploits the affine redundancy that is commonly present in most images; this redundancy is related to the similarity of an image with itself. As shown in Fig.1, part A of a certain image is similar to another part B of the image [1, 2].

Fig.1: Affine redundancy presents in Peppers image.

Fractal image compression finds similar patterns that exist on different scales and at different places in an image, and then eliminates as much redundancy as possible. Images contain lots of repetition such as one piece of grass, one leaf, one edge, one bit of blue sky, one eye and so on. Any fractal compression system involves a process that looks for these repetitions across scale and position. Then, instead of saving information about every similar part of an image, the system merely saves the common representation and how it is used on different scales and in different places [3].

II. ENCODING PROCESS

Encoding mechanism of fractal technique is different from all encoding methods. In fractal encoding, the image is compressed by using the partial self-similarity of the images as redundancy, self-similarity well approximating the block to be encoded is extracted from the image and the transform parameter for a contraction transform representing the self-similarity is used as code [4]. Therefore, an image is partitioned into a set of ranges blocks. The encoder has to solve the following problem:

$$R \cong sD + oI \dots\dots (1)$$

For each range block R the best approximation needs to be found, where D is a codebook block transformed from a domain block to the same size as R. The coefficients s and o are called scaling and offset.

The encoding of each range block search through all of Domain pool (D) to find a best domain D_i (i.e. $D_i \in D$) which minimizes the collage error:

$$E(R,D) = \|R - sD - o\|^2 \dots\dots (2)$$

That is, find the part of image that most looks like the image above R_i. There are 8 ways to map one block onto another. Minimizing eq. (1) means that: First, it means finding a good choice for D_i, second, it means finding good contrast s_i and brightness o_i for w_i [5, 6].

III. HV- Partitioning

Horizontal-Vertical (H-V) partitioning, an image is partitioned either horizontally or vertically to form two new rectangular, like the quadtree, the partitioning repeats recursively until a covering tolerance is satisfied, see Fig.2 [5, 7].

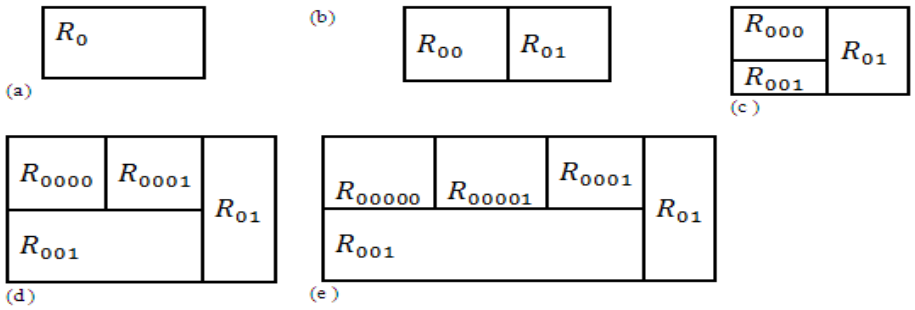


Fig.2: H-V partitioning process a) original image, b) 1st partition: partitioning the image vertically into two parts, c) 2nd partition: partitioning the first part horizontally into two parts, d) 3rd partition: partitioning the first part vertically into two parts, and e) 4th partition: partitioning the first part vertically into two parts.

In HV-Partitioning when a D_i for an R_i whose approximation does not match a specific expectation, then divide R_i into two blocks (horizontally or vertically), see Fig.2, that do not have to be of equal area. In this way can be observed self-similarities in the image and choose the divisions wisely. HV-Partitioning will result in fewer numbers of quadrilaterals, which means a set R of smaller size [8].

IV. DECODING IMAGE

The most remarkable of fractal image compression is the simplicity of the decoder. The reconstruction of the image starts with any arbitrary image even a blank image, of the same size as the original. This is the step 0 image, which used to construct the step 1 image. A step 1 image then construct by applying the transformation of each domain block to the range block taken from the step 0 image. The range block is found by taking the same sub-section of the step 0 image that the best-match range block occupied in the original image [9]. The approximate step 1 image will be slightly closer to the image. In the same way, a step 2 image is constructed from the step 1 image and so on. This process is repeated until the step n image and the step n-1 image are indistinguishable, and repeating the process will add no new detail. This image is called the fractal attractor of the transformations, and it approximated the original image [1].

V. FIDELITY CRITERIA

By taking the advantage of the redundancy that exists in the image can be determine the minimal data required retaining the necessary information and that it is the key of image compression [10]. Removal of psych-visually redundant data results in a loss of real or quantitative visual information. Thus, the information losses, and usually associated with compressed image which may be acceptable or not depending on some predefined tolerated level. Therefore, the fidelity criteria are often used to measure the amount of information losses [8].

In our paper, the analysis of the results depend on the objective fidelity criteria because this class of criteria is simple to calculate and convenient to having more accurate results. The SNR and PSNR are often measured on a logarithmic scale and the units of measurement are called decibels (dB) since [11]:

$$MSE = \frac{1}{NXN}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[f^{\wedge}(x,y) - f(x,y)]^2 \ldots\ldots(3)$$

Where: $f^{\wedge}(x,y) and\ f(x,y)$: The reconstructed and original image respectively.

$$SNR = \left[\frac{f^{\wedge}(x,y) - f(x,y)}{MSE}\right] \ldots\ldots(4)$$

$$PSNR = 10log_{10}\frac{(l-1)^2}{MSE} \dots (5)$$

Where L: Is the gray scale of the image.

VI. FRACTAL COLOR IMAGE COMPRESSION DESIGN

Fractal method to encode a color images by gray-level image encoding algorithm. RGB color image must be split into three channels red, green, and blue, then compressed them separately by treatment each color component as a single gray-scale image. Three channels are partitioned, encoded, and decoded separately. Obviously it is time consuming, since if the spent time for encoding a gray level image is *t* then, the expected time required to encode the RGB color image is 3*t*. In this paper we are design the model of fractal RGB-image compression system with discussing the two implemented ways of H-V partitioning schemas.

A pixel of a color image contains of three bytes represents the three channels of color, Fig.3. So RGB image convert into three arrays, one of each Red, Green, and Blue. Each one of these arrays refers to as *range* and will feed to the next module in the system (i.e. partitioning module).

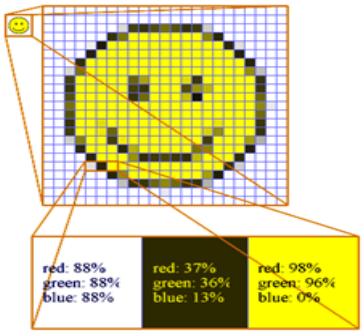
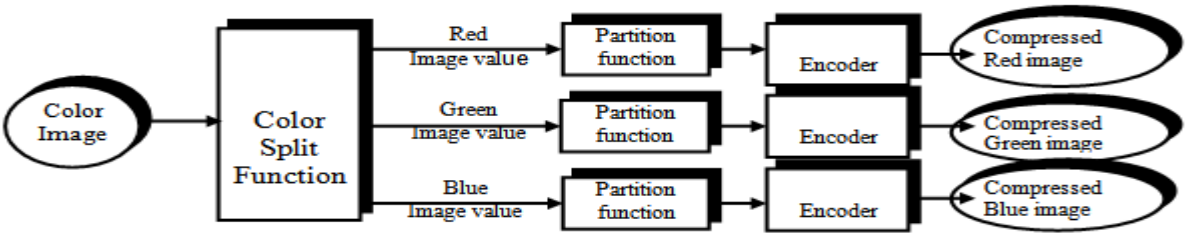


Fig.3: Every square represents a pixel whose colors are constructed by adding the values for red, green and blue.

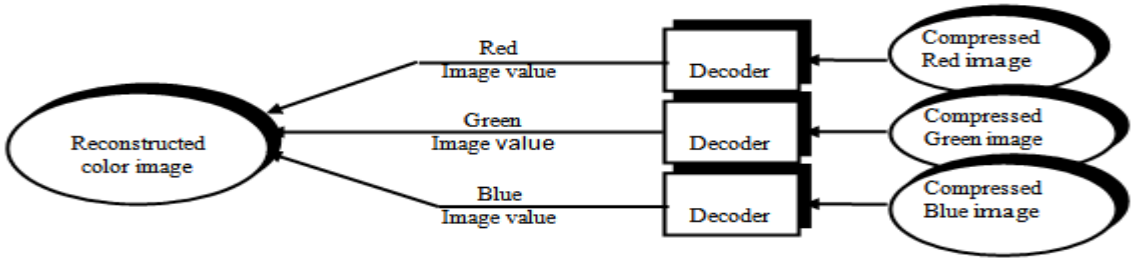
Any RGB color pixel handling as:

I. By using AND operation with value 255, get the first byte, (red color value).
II. By using, 8 bit shift operation, AND operation with value 255, get the second byte, (Green color value).
III. By using, 16 bit shift operation, AND operation with value 255, get the third byte, (Blue color value).

The design of encoder and decoder of fractal RGB image are illustrated in Fig.4.



a) Encoder



b) Decoder

Fig.4: The block diagram of fractal RGB compression.

IV. JOINTLY AND DIFFERENT PARTITIONING CHANNELS

Many ways to partition the image are known, many researchers and their experiments shows that the horizontal and vertical (H-V) partitioning is the best way for reducing the encoding time with best quality of reconstructed image. Therefore, H-V partitioning used in our work.

H-V partitioning divide the image either horizontally or vertically to yield two new rectangles instead of quarter in the quadtree partitioning case, therefore, fewer number of range blocks are produced. Algorithm1 state the required steps of H-V partitioning.

Algorithm 1: H-V partitioning steps.

Step1: Read the initial image and store its values in an array.
Step2: Compute the global mean of initial image.
Step3: Compute the global standard deviation of the initial image:
Step4: Set inclusion factor value (I_f), and acceptance ratio (A_R) value.
Step5: Compute the extended standard deviation (ESD) of each image by using the IncF.
$ESD = \sigma_D \times I_f \dots \dots (6)$
Step6: Set maximum and minimum block size value.
Step7: Set range index to zero.
Step8: Set item-No. to 1.
Step9: Initialize array of record to store information about each block obtains from HV partitioning process, each record consists of the following fields:

1. Top-left position coordinates (X,Y) of the image block.
2. Width of the image blocks (X-size).
3. High of the image block (Y-size).
4. Pointer of the image block (Next).

Step10: Compute the mean of a range block.
Step11: For each block of range, compute the average of abnormal pixel. First, compute the number of pixels whose values differ from the mean by a value greater than the extended standard deviation, then compare their average with the selected ratio value to decide either partition or not partitioning that block.
Step12: Set Flag to" ".
Step13: If the average of abnormal pixel is greater than the acceptance ratio then, determine the location of the greatest number of abnormal pixels if they are on the top or bottom, in the right or left by make a comparison between the numbers of abnormal pixel in each orientation. Then, take the large value of the two comparisons as below:

1. if SL>SR then X-direction = left else X_direction = right
2. If ST>SB then Y-direction = top else Y_ direction = bottom

Where:
SL is the summation of the abnormal pixel on the left.
SR is the summation of the abnormal pixel on the right.
ST is the summation of the abnormal pixel on the top.
SB is the summation of the abnormal pixel on the bottom.

3. If X_direction >= Y_direction then set flag = "H"(i.e. horizontal partition)
else set flag = "V" (i.e. vertical partition).

Step14: From step 12, If the average of abnormal pixel is less than the acceptance ratio (AR), there is no partition.
Step15: Repeat steps 10-14 until carrying out the minimum size of range blocks

In our proposed paper we have two ways to partition RGB-image:

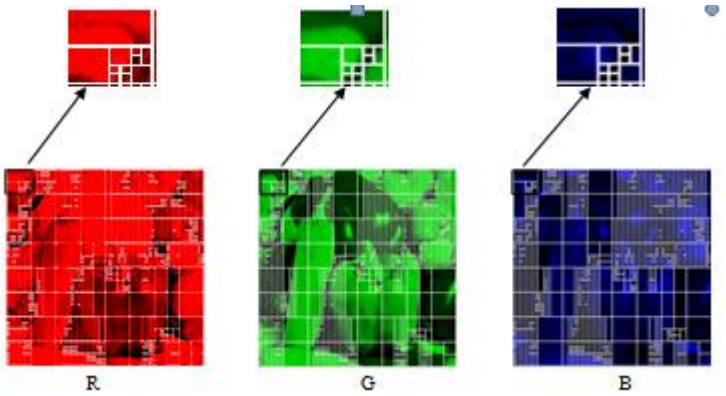
1. Jointly Partitioning RGB Image

Jointly partitioning schema is partitioned RGB image (i.e. all three channels Red, Green, and Blue) at one time depending on the information regions of image by using threshold value of partitioning. In this schema, three channels have the same number of range blocks and the same range coordinates, Fig.5a demonstrate this partitioning idea. The block diagram of encoder, Fig.4a, is modified, and Fig. 6 shows the modification. After jointly partitioning, the RGB image channels are prepare to the next stage (i.e. encoding stage).

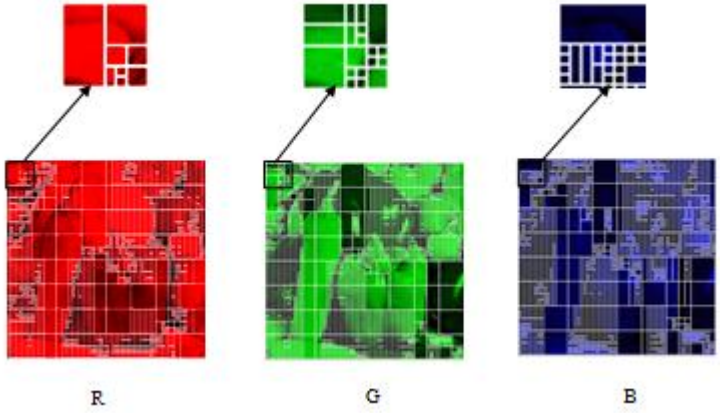
2. Different Partitioning RGB Image

Jointly partitioning operation has a weak point, because whole image channels have the same range blocks. In this schema, each channel will partition depending on its information, and then we have different range blocks for each Red, Green, and Blue channel (see Fig.5b). Different partitioning idea is more effectively

in encoding stage, because each channel has the different number of ranges and each region will cover well. Fig.5b shows the mechanism of the different partitioning idea.



(a)



(b)

Fig.5: H-V partitioning idea by using: (a): joint partition (b): different partition

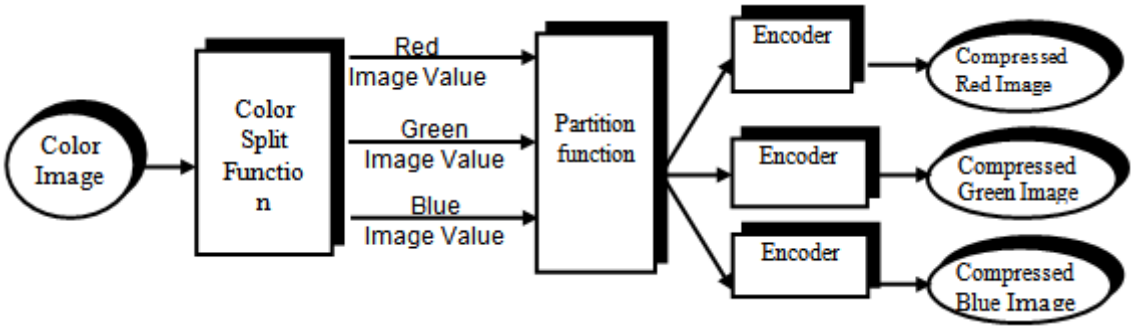


Fig.6: The block diagram of the RGB encoder using Jointly Partition way

V. ENCODER MODULES

Encoder module consists of many subs modules work together to compressed image file. These modules are:

I. Domain Creating Module

The domain is another array, which it differs from the range array in size. Stepping through the domain array horizontally and vertically will lead to create a list of domain blocks. The number of domain blocks depending on the step size.

In our work the average is taken of every four (2 x 2) adjacent element in range to be one pixel in the domain. Fig.7a illustrates the average method to construct the domain block and Fig.7b shows an example of range and domain size. The entire image is reduced at the start of the compression operations, and the domain blocks are then chosen from the reduced image with no need for further scaling. This way of creating the

domain has two important points; it greatly decreases the computational process of the program, but reduced the number of domain blocks by a factor of four.

II. Mapping Module

Each range block from range pool will be mapped to all same size domain block in domain pool. The first block in horizontally and vertically direction will be first start. A domain pool may contain of overlapped or non-overlapped domain block according to the step size, if the step size is equal to the block size, the domain pool will be without overlap, and if the step size is less to the block size then domain pool have overlapping. Also each range block compared with domain blocks in eight symmetries and this may lead to best approximation (i.e. less error) but result in more computational time to perform an extra matching process. Mapping operation done by searching in domain pool to find domain block d_k satisfying the best match with the range block under search. Best matching can be computed by minimized the error between range and domain blocks. Algorithm2 state the required steps of mapping operation.

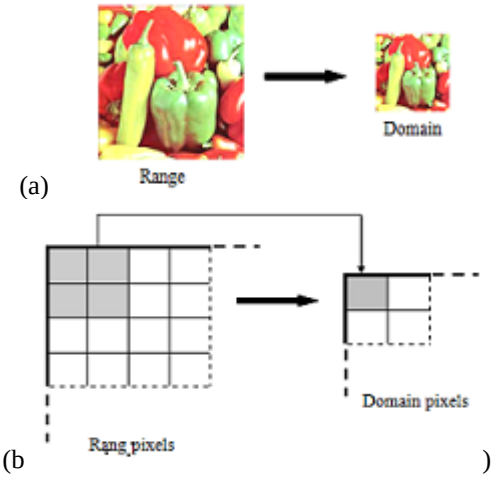


Fig.7: Construction domain block from the range block; a): Create the domain block from the range block, every four adjacent pixel in the range image are combined to produce one pixel in the domain, b) Range and Domain size

Algorithm2: Mapping operation steps.

I) Compute the sum of square error according to following equation:

$$E(R,D) = \frac{1}{n}\left[\sum_{i=1}^{n} r_i^2 + s\left(s\sum_{i=1}^{n} d_i^2 - 2\sum_{i1}^{n} d_i r_i + 2o\sum_{i=1}^{n} d_i + o(no - 2\sum_{i=1}^{n} r_i\right)\right] \dots (7)$$

Where n: the no. of pixel in each block, r_i: the range pixel value, d_i: the domain pixel value, s: scale factor, and o: offset factor.

s and o are also called the IFS coefficients , and determined by:

$$s = \frac{n(\sum_{i=1}^{n} d_i r_i) - (\sum_{i=1}^{n} d_i)(\sum_{i=1}^{n} r_i)}{n\sum_{i=1}^{n} {d_i}^2 - (\sum_{i=1}^{n} d_i)^2} \dots\dots (8)$$

$$o = \frac{\sum_{i=1}^{n} r_i \sum_{i=1}^{n} {d_i}^2 - \sum_{i=1}^{n} d_i \sum_{i=1}^{n} d_i r_i}{n\sum_{i=1}^{n} {d_i}^2 - (\sum_{i=1}^{n} d_i)^2} \dots\dots (9)$$

II) IFS coefficients have a limited interval of values, the scale interval is [-1,1], and the offset interval is [-256, +256], any value not in the interval will regardless, as in the steps:

Step1. In previous statement, initialize the minimum and maximum s and o values

Step2. If s (which calculated from matching between R and D) > maximum value then s = maximum value
Else if s < minimum value then s = minimum value.

Step3. If o (which calculated from matching between R and D) > maximum value then o = maximum value
Else if o< minimum value then o = minimum value.

3. Quantization Module

The IFS coefficients (*s* and *o*) are real values, in order to increase the compression ratio; these values must be quantized before saving. Quantization is the process that reduces the number of bits needed to store the coefficient values by reducing its precision from float type to integer. This process may be performed either by using uniform or non-uniform method. In our work used the uniform quantization. Algorithm3 show uniform quantization.

Algorithm3: Uniform quantization steps

Step1: Set b_s and b_o values, the number of bits assigned to scaling and offset respectively.

Step2: Compute the *step s* and *step o* values by the following equations:

$$Step\ s = \frac{(2^{bs} - 2)}{MaxScl - MinScl} \dots\dots (10)$$

$$Step\ o = \frac{(2^{bo} - 2)}{MaxOfs - MinOfs} \dots\dots (11)$$

Step3: For all s and o values perform the quantization by using the uniform quantize eqs 11and 12:

$$S_q = round\big(step\ s \times (s - MinScl)\big) \dots (12)$$

$$O_q = round\big(step\ o \times (o - MinOfs)\big) \dots (13)$$

Where: s_q and $o_{q\ are}$ represented the quantize values of s and o respectively.

4. Minimization Coordinates

The encoding parameters are storing in form of bits and the compression ratio depending on the sum of these bits, therefore, a good matter is to find a way to reduce these bits to increase the compression ratio. A simply reducing done by dividing the value of X and Y (domain coordinate), which are an integer values, by a factor (certainly multiple of 2). More suitable and present factor is the domain step value itself.

5. Code Generation (IFS code)

The information of good approximate of range-domain <R, D> (X, Y, s, o, Symmetry) resulted from Encoder modules, saved in array of record; the size of array is equal to the number of range blocks. Table1 illustrates this information.

Table1: IFS mapping information, which saved in array of record.

Parameter	Description	No. of bits
X , Y	The coordinate of the best matched domain block	5
s	The scale value of the best matched domain block	5
o	The offset value of the best matched domain block	7
Symmetry	The state (domain orientation) that gives good approximation	3

VI. IMPLEMENTATION AND RESULTS DISCUSSION

In our work, there are many of control parameters are studied and analyzed their behavior. It is important to realize how computationally intensive the fractal encoding of color image is:

a. Fractal RGB-Image Compression based on Different Partitioning RGB Image (DP-F–RGB–C)

The fractal RGB-image compression was implemented on Peppers RGB image of size 256 x 256 by using Different Partitioning ways Tables2 and 3 demonstrate the results based on varying the value of A_R and fixing I_f and vice versa.

We found that when the maximum block size=8 and minimum block size=4, the best values of (A_R ranges between (0.1-0.7). Also the results have indicated that the selection of small values of Inclusion Factor or Acceptance Ratio had produced high image quality, low compression ratio, long encoding time, and vice versa. From the table it is obvious that there is a relationship between CR and PSNR at different value of Inclusion Factor or Acceptance Ratio. A higher compression may be gained but the distortion level will increase. When the number of range block small we get high value of C.R., this value decrease when number of range block increase exponentially. Small number of block gives low values of SNR and PSNR, and so, increase number of ranges blocks lead to high value of SNR and PSNR. Since there are more range blocks, then more time required for mapping search, therefor increase number of ranges blocks lead to more encoding time. Fig.8 showed the behavior of encoding time with number of range block.

Table2: Fractal encoding and decoding results using HV partitioning method applied on the Peppers RGB color image with maximum block size=8 and minimum block size=4.

Domain Step=4 s = 5 bits , o = 7 bits								Minimum Scale = -1.2 Maximum Scale =1.2 Minimum Offset = -255 Maximum Offset =255						
I_f	A_R	No. of Blocks			Comp. Ratio			SNR(dB)			PSNR(dB)			Time (Sec)
		R	G	B	R	G	B	R	G	B	R	G	B	
0.3	0.1	2956	2627	3035	6.40	7.26	6.33	28.80	28.15	24.16	32.88	33.51	34.02	81
	0.3	2577	2323	2668	7.41	8.19	7.17	28.41	27.63	23.73	32.50	33.00	33.60	77
	0.5	2160	2095	2363	8.80	9.04	8.06	27.62	26.82	23.08	31.71	32.16	32.94	72
	0.7	1781	1854	1983	10.60	10.18	9.55	26.53	25.88	21.88	30.62	31.25	31.76	68
0.5	0.1	2519	2301	2623	7.56	8.26	7.28	28.37	27.55	23.63	32.64	32.91	33.49	76
	0.3	2069	1992	2218	9.16	9.49	8.57	27.48	26.53	22.75	31.56	31.90	32.62	71
	0.5	1727	1761	1889	10.96	10.70	10.00	26.40	25.49	21.63	30.49	30.86	31.51	66
	0.7	1361	1530	1514	13.81	12.28	12.43	25.18	24.29	20.23	29.27	29.67	30.12	62
0.7	0.1	2229	2096	2276	8.51	9.03	8.00	27.79	26.81	23.17	31.88	32.17	33.04	71
	0.3	1743	1733	1891	10.82	10.86	9.99	26.54	25.35	21.74	30.63	30.73	31.61	66
	0.5	1368	1534	1534	13.75	12.24	12.26	25.23	24.48	20.45	29.32	29.86	30.34	63
	0.7	1169	1354	1235	16.05	13.84	15.19	24.37	23.46	19.20	28.46	28.84	29.10	60

Table3: Fractal encoding and decoding results using HV partitioning method applied on the Birds RGB color image with maximum block size=8 and minimum block size=4.

Domain Step=4 s = 5 bits , o = 7 bits								Minimum Scale = -1.2 Maximum Scale =1.2 Minimum Offset = -255 Maximum Offset =255						
I_f	A_R	No. of Blocks			Comp. Ratio			SNR(dB)			PSNR(dB)			Time (Sec)
		R	G	B	R	G	B	R	G	B	R	G	B	
0.1	0.1	3542	3696	3719	5.45	5.23	5.20	25.95	25.70	23.05	32.34	32.27	32.50	90
	0.3	3108	3195	3364	6.18	6.02	5.73	25.87	25.63	22.92	32.25	32.20	32.88	85
0.3	0.1	2359	2465	2533	8.07	7.73	7.53	25.67	25.46	22.58	32.06	32.03	32.04	75
	0.3	2034	2115	2126	9.32	8.96	8.93	25.30	25.23	22.23	31.64	31.81	31.69	70
	0.5	1761	1815	1835	10.72	10.40	10.31	24.54	24.80	21.81	30.93	31.37	31.27	67
0.5	0.1	1943	2060	2059	9.50	9.19	9.22	25.33	25.26	22.22	31.72	31.84	31.68	70
	0.3	1665	1664	1721	11.32	11.33	10.92	24.88	24.63	21.63	30.67	31.21	31.09	64
	0.5	1537	1472	1519	12.23	12.76	12.38	23.89	24.13	21.06	30.29	30.71	30.54	63

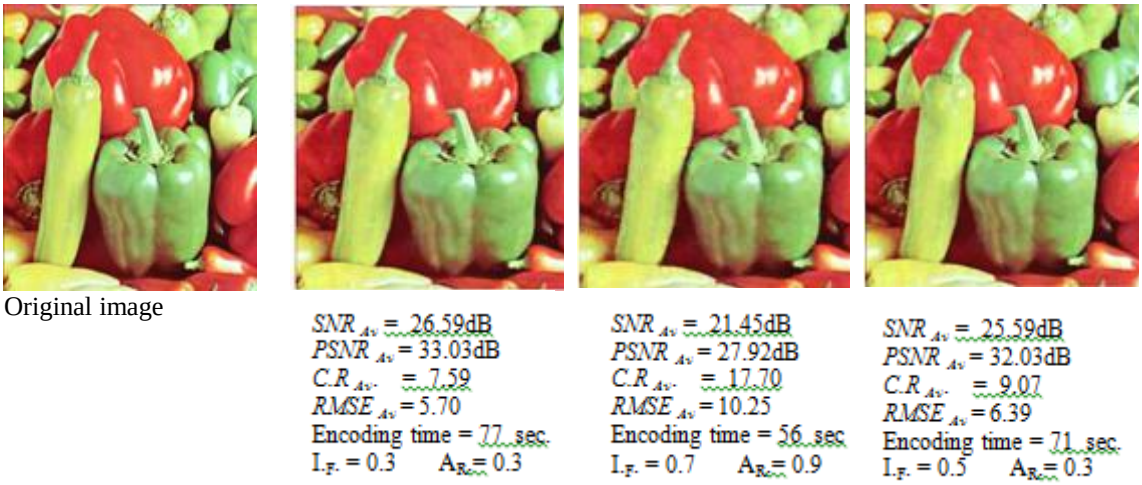


Fig.8: RGB color image reconstruct under multi control parameters.

b. Fractal RGB-Image Compression based on Jointly Partitioning (JP-F–RGB–C)

Implementation of this schema likes the previous one. Table4 illustrate some result gets from this schema. The same Inclusion factor and ratio values of the previous schema will not give the same partitions number as before, sine this schema produce equal number of ranges with the same structured (i.e. same coordinates) for all the three band (channel) of color, see Fig.5 again. Fig.9 shows some reconstructed image results from this schema of compression.

Table4: The Jointly Partition Fractal encoding and decoding results using HV partitioning method applied on the Peppers RGB color image.

Domain Step=4 S = 5 , O = 7 Maximum Block Size= 8 Minimum Block Size =4				Minimum Scale =-1 Maximum Scale =1 Minimum Offset = -256 Maximum Offset =256		
I_f	R_A	No. of Blocks R = G = B	Average C. R	Average SNR(dB)	Average PSNR(dB)	Time (Sec.)
0.3	0.1	2406	8.45	26.33	32.77	74
0.3	0.3	2022	10.04	25.21	31.67	70
0.3	0.5	1639	12.63	23.83	30.28	64
0.3	0.7	1274	15.89	22.41	28.87	59
0.3	0.9	1074	18.83	21.48	27.95	56
0.5	0.2	1794	11.30	24.43	30.88	67
0.5	0.5	1282	15.79	22.32	28.78	59
0.7	0.3	1341	15.09	22.45	28.91	60
0.7	0.7	1064	19.01	21.48	27.94	56

Fig.9: RGB color image reconstructed under multi control parameters,

VII. COMPARISON

A comparison between two suggested schemas are depended on the same number of range blocks or near too. Obviously, there are no noticeable changes between the two schemes except the compression ratio,

which is because the partitioning data, since three sets of partitioning data have to be saved in the DP-F–RGB–C, (each color band has its special partitions). But, there is only one set in case of JP-F–RGB–C; (the same partitions for all bands) see Table5.

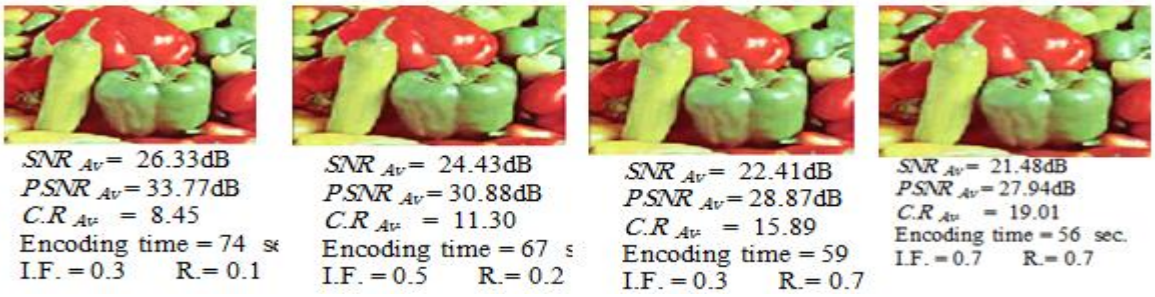


Table5: A comparison between the DP-F–RGB–C schema and the JP-F–RGB–C schema when maximum block size = 8, minimum block size = 4.

Schema type	Average number of range block	Average C. R	Average SNR(dB)	Average PSNR(dB)	Time (Sec.)
DF–RGB–C	1789	10.56	24.54	30.99	66
JP F–RGB–C	1781	11.30	24.35	30.79	66
DF–RGB–C	1058	17.70	21.45	27.92	56
JP F–RGB–C	1060	19.01	21.47	27.94	56

VIII. CONCLUSION

From the results of these tests, we conclude that by reduction Domain pool , the PSNR and the encoding time are directly proportional with the domain pool size, while the compression ratio has inversely proportional. This method speeds up the encoding time about 60%.

REFERENCES

[1]. Jamila H. S., "Fractal Image compression", Ph.D. thesis, College of Science, University of Baghdad, January,2001.

[2]. Jacquin A.E., "A fractal theory of iterated Markov operators with applications to digital forms" in proceeding from SPIE visual communication and image processing, Vol.1360, 1990

[3]. Robert J., "Combining Fractal Compression with Sampling and Interpolation", Internet paper, January 10, 2003.

[4]. Ryuji F. and Ma Sahino W., "Fractal Image Coding with a Multi scaling- Domain" Electronic and communication in Japan, part 3 , vol.87, No.2,2004.

[5]. Fisher, Y 1995, "Fractal Image Compression: Theory and Application", Springer-Verlag New York, Inc., New York.

[6]. Hannes H. and Dietmaris. , "VQ- Encoding of Luminance Parameters in fractal coding schemes", ICASSP-97, Apr.21-24, Munich, Germany,1997

[7]. Saupe , D. Hamzaoui, R., "Fractal Image Compression An Introduction Over View" Lasagna. 1996

[8]. Ghada K., "Adaptive Fractal Image Compression", M.Sc. thesis, National computer Center /Higher Education Institute of computer and Informatics, 2001.

[9]. Gordon W. Paynter "Fractal Image Compression" 0657.420 of an investigation, Department of computer science, University of Waikato, Hamilton, New Zealand, 1995.

[10]. Scott, E. U., "Computer Vision and Image Processing Practical Approach using CIVP tools ", Prentice-hall, Inc., USA, 1998.

[11]. Gonzalez R. C., and Wintz, P., "Digital image processing", 2nd ed., Addison-Wesley Publication Company, 1987.