Journal of Al-Nahrain University Science **Vol.15 (4), December, 2012, pp.٢٠٠-٢٠٧**

The Use of Double Moment Based Descriptors to Speed up FIC

Suhad L. Mahmoud
Department of Computer, College of Science, Al-Nahrain University, Baghdad-Iraq.
E-mail: suhadlatef@yahoo.com.

Abstract

In this research, an improved Partitioned Iterated Function System (PIFS) scheme is introduced to speed up the encoding stage time in Fractal Image Compression (FIC). The proposed system is based on using both symmetry prediction and blocks indexing to speed up the block matching process; these two methods have been used to manage the way of selecting the suitable domain to represent the range block. For block indexing the moment features have been used for determining double block descriptors, each one is affine invariant. The two block descriptors are utilized in a combined indexing scheme to classify the blocks of both range and domain pool. So, the block indexing method is introduced to filter the domain blocks, and keeps only those domain blocks have similar block indices with that of the mapped range block which will be approximately represented using affine mapping. The symmetry predictor is used to reduce the number of isometric trails from 8 to one trail .The research also includes the implementation of the use of single moment descriptor to speed up FIC for comparison. The test results indicated that the proposed improvement has reduced the required encoding time to 0.25 second, and the attain compression ratio is 7.99 without making significant degradation in image quality.

Keywords: IFS, Fractal Image Compression, Isometric Processes, Image compression.

Introduction

Various compression methods have been proposed to achieve high compression ratios and high image qualities in low computation time. Among the image compression methods, the fractal image coding method based on the theory of iterated function system (IFS) has captured increasing attention and interest. The application of fractal models to image compression has been prompted by Barnsley [1,2]. The first automated fractal coding algorithm based on Partitioned (local) Iterated Function System (PIFS) was developed by Jacquin [3,4].

The basic idea of fractal image compression is the partitioning of input image into non-overlapping range blocks. For every range block a similar but larger domain block is found. The set of coefficients of mapping the domain blocks to the range block, using affine transform, is recorded as compression data. The compressed image data set is called the Iterated Function System (IFS) mapping set. Decoding process applies the determined IFS transformations on any initial image, and the process is repeated many times till reaching the attractor.The main problem in the fractal image compression (FIC) method is the long encoding time. Several researches had been introduced that proposed improvements on PIFS scheme to speed up the encoding time. Some proposed methods involves combination of fractal coding; either using cosine transform (DCT) [5], or Wavelet transform [6]. Other coding methods are based on using moment descriptor as a criteria to classify the range-domain blocks [7,8].

Proposed System

In the proposed system, the RGB color image is transformed into (Y, Cb, Cr) color representation, then the chromatic bands (i.e. Cb and Cr) have been down sampled (by 2) to reduce the encoding time and to improve the compression gain [9]. Each band is considered as a range pool and down sampled (by 2) to produce the domain pool. The range pool is partitioned into n×n non-overlapped blocks, while the domain pool is partitioned into n × n overlapped blocks.

The IFS with symmetry prediction and double descriptors have been used. These descriptors are based on the values of central moment's order-1 and order-3, they have been combined to produce a block descriptors characterized as affine transform invariant

descriptors to classify the domain and range blocks into classes to speed up the IFS matching. The isometric predictor; which is based on the moment status (order-1) of the matched block is utilized to reduce the matching time because only one match, instead of eight, will be tested to evaluate the IFS-similarity between any range and domain block[7]. To achieve high compression ratio, the coefficients of the optimal affine approximation are quantized. The quantized affine transformation coefficients of the image are stored as a compression file.

Block Moments

In general**,** moments are set of parameters which describes the distribution of material (in image processing it is equivalent to brightness) relative to a reference point or an axis. The idea of using moments to construct the image feature vectors is one of the most common methods used today. Each moment order reflects different information for the same image. For a 2-D continuous function *f*(x,y), the moment of order (p+q) is defined as [10,11]:

$$M(p,q) = \sum_{p}\sum_{q} x^{p} y^{q} I(x,y) \quad \text{..... (1)}$$

The pixel whose coordinates (x=0 or y=0) yield (0) in the moment equation. But the central moment for a function I(x, y) can be expressed as [3, 4]:

$$M(p,q) = \sum_{p}\sum_{q} (x-x_c)^{p}(y-y_c)^{q} I(x,y) \quad \text{..... (2)}$$

$$x_c = \frac{M_{10}}{M_{00}} \quad \text{..... (3)}$$

$$y_c = \frac{M_{01}}{M_{00}} \quad \text{..... (4)}$$

Where:

x_c is the x coordinate of the center point.

y_c is the y coordinate of the center point.

M_{00} is the zero order moment.

M_{10} is the first order moment relative to x axis.

M_{01} is the first order moment relative to y axis.

For an image block I(x, y) {x, y | 0, 1, …., m-1}, its first and third order centralized moments are defined as :

$$M_{10} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(x-c) \quad \text{..... (5)}$$

$$M_{01} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(y-c) \quad \text{..... (6)}$$

$$M_{30} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(x-c)^{3} \quad \text{..... (7)}$$

$$M_{03} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(y-c)^{3} \quad \text{..... (8)}$$

IFS Mapping

Suppose that each range block have pixel intensities (r_1,..,r_n) and each domain block have pixel intensities (d_1,..,d_n). Then for each range block (r_i) the optimal affine approximation can be represented as [8,12].

$$r' = s(d_i - \bar{d}) + \bar{r} \quad \text{..... (9)}$$

where

r' is the optimal approximated i^{th} pixel value in the range block.

d_i is the corresponding pixel value in the domain block.

s is the scaling coefficient.

$\bar{r}, \bar{d}$ are the average of the range block and domain block, respectively.

The search process implies that all domain blocks (d_i) listed in domain pool should be matched with the considered range block, to find out the optimal approximation (r_i').

After the generation of the range and domain pools; each range block listed in range pool is taken separately and approximated by one of the domain blocks listed in domain pool. At each approximation instance, the mapping coefficients,(i.e. scale coefficients) and range average $\bar{r}$, are determined. These coefficients are called the affine coefficients. The value of scale (s) parameter is determined by finding the least sum χ^2 of square error between r_i' and r_i according to following equations.

$$\chi^2 = \sum_{i=0}^{m-1}\left[(r_i - \bar{r}) - s(d_i - \bar{d})\right]^2 \quad \text{..... (10)}$$

The minimum of χ^2 occurs when:

$$\frac{\partial \chi^2}{\partial s} = 0 \quad \text{..... (11)}$$

Combining equations (9), (10) leads to:

$$s = \begin{cases} \dfrac{\frac{1}{m}\sum_{i=0}^{m-1} d_i r_i - \bar{r}\bar{d}}{\sigma_d^2} & if\ \sigma_d^2 > 0 \\ 0 & if\ \sigma_d^2 = 0 \end{cases} \quad \text{..... (12)}$$

Where,

$$\sigma_d{}^2 = \frac{1}{m}\sum_{i=0}^{m-1} d_i{}^2 - \bar{d}^2 \quad \text{..... (13)}$$

r_i is the pixel value of the range block.

m is the number of pixels in each block (the block size)

At each range-domain matching instance, and before determination of χ^2, the following condition should be applied on the determined values of s:

$$s = \begin{cases} -s_{\max} & \text{if } s < -s_{\max} \\ s & \text{if } -s_{\max} \le s \le s_{\max} \\ s_{\max} & \text{if } s > s_{\max} \end{cases} \quad \text{..... (14)}$$

And then, it should be quantized using the following equations:

$$I_s = round(\tfrac{s}{Q}) \quad \text{..... (15)}$$

$$\tilde{S} = Q_s I_s \quad \text{..... (16)}$$

$$Q_s = \tfrac{S_{\max}}{2^{bs-1}-1} \quad \text{..... (17)}$$

Where

S_{max} is the highest permissible value of the scale coefficients.

Q_s is the quantization step of the scale coefficients.

I_s is the quantization index of the scale coefficients.

bs is the number of scale bits.

The quantized value of scale (s) is used to calculate the sum of square error χ^2 using equation (10)

Symmetry Process

In order to increase the size of domain pool and, consequently, to increase the probability of finding the best (near optimal) approximations for the range blocks, each domain block is transformed using a set of isometric transforms (i.e., rotation and flipping), such that eight versions (blocks) are produced for each domain block [13]. The eight isometric mappings are (identity, rotation 90, rotation 180, rotation 270, reflection-x, reflection with rotation 90, reflection with rotation 180, reflection with rotation 270), The number of bits required to represent the eight isometric mapping cases is (3) bits.

The main disadvantage of using the isometric mappings in the encoding process is that more computational time is required to perform the extra matching processes. Therefore in our proposed system a symmetry predictor was added to reduce the number of isometric trails from 8 to one trail [7]. The first order centralized moments (equations 5 & 6) are used to index each domain and range blocks onto one of eight possible isometric states, the criteria used to context the isometric state indexing is shown in Table (1). At each range-domain matching instance, the indices of both domain and range blocks are passed through the predictor, see Table (2), then the predictor outputs the index of the required isometric transform to get the best possible match between the domain and range blocks .

Table (1)
The Truth Table For The Eight Blocks States[7].

Block's Class Index	***Boolean Criteria***		
	$\|M_{10}\| \ge \|M_{01}\|$	$\|M_{10}\| \ge 0$	$\|M_{01}\| \ge 0$
0	T	T	T
1	T	T	F
2	T	F	T
3	T	F	F
4	F	T	T
5	F	T	F
6	F	F	T
7	F	F	F
0≡ No Operation; 1≡ Rotation_90; 2≡Rotation_180; 3≡Rotation_270; 4≡Reflection; 5≡Reflection+Rotation_90; 6≡Reflection+Rotation_180; 7≡ Reflection+Rotation_270;			

Table (2)
The Required Isometric Operation to Convert the Block State [7].

		Domain Block Index							
		0	***1***	***2***	***3***	***4***	***5***	***6***	***7***
Range Block Index	*0*	0	6	4	2	5	3	1	7
	1	6	0	2	4	1	7	5	3
	2	4	2	0	6	3	5	7	1
	3	2	4	6	0	7	1	3	5
	4	5	1	3	7	0	4	6	2
	5	3	7	5	1	4	0	2	6
	6	1	5	7	3	6	2	0	4
	7	7	3	1	5	2	6	4	0

0≡ No Operation; 1≡ Rotation_90;
2≡ Rotation_180;
3≡ Rotation_270; 4≡ Reflection;
5≡ Reflection+Rotation_90;
6≡ Reflection+Rotation_180;
7≡ Reflection+Rotation_270;

Encoding Process

The method applied to speed up the encoding of range blocks using double moment descriptor instead of single descriptor are summarized by the following steps:

1. Load the (R, G, B) image (i.e., three 2D arrays).
2. Convert (R,G,B) arrays to (Y,Cb,Cr) arrays
3. Down sample Cb and Cr to quarter of its original size.
4. For each component (i.e., the original Y, and the down sample Cb and Cr) establish the range image array. The range array must be partitioned into non-overlapping fixed blocks, to generate the range blocks ($r_1,\ldots,r_n$).
5. Down sample the original Y, the down sampled Cb, and the down sampled Cr by 2 to get ($Y^{'}$,$Cb^{'}$, and $Cr^{'}$).
6. For each component (i.e. $Y^{'}$,$Cb^{'}$, and $Cr^{'}$) establish the domain image (array). The domain must be partitioned into overlapping blocks, to generate the domain blocks ($d_1,\ldots,d_n$). They should have the same size of range blocks.
7. For each domain block do the following:
   a. Calculate the average ($\bar{d}$).
   b. Calculate the moments $M_d(0,1)$, $M_d(1,0)$, $M_d(0,3)$, and $M_d(3,0)$.
   c. Determine the block isometric index sym_indx_d that based on moment order-1 (Table (1)).
   d. Determine moment ratios (R_{1d}) and (R_{3d}) depending on Moment order-1 and order-3 respectively :

$$R_{1d} = \begin{cases} \frac{M(1,0)}{M(0,1)} \times N_m & if\ M(0,1) > M(1,0) \\ \frac{M(0,1)}{M(1,0)} \times N_m & if\ M(1,0) > M(0,1) \end{cases} \quad \text{..... (18)}$$

$$R_{3d} = \frac{M^2(3,0) - M^2(0,3)}{M^2(3,0) + M^2(0,3)} \times N_m \quad \text{..... (19)}$$

   e. Determine the moment index value Ix_d as linear combination of two descriptors (R_{1d}) and (R_{3d}) using the following equation:

$$Ix_d = R1_d \times (N_m + 1) + R3_d \quad \text{..... (20)}$$

Where

N_m is the maximum moment index value.

So the number of classes $= N_m \times (N_m+1)+N_m$

The index Ix_d is used to classify domain blocks. (i.e. each class includes blocks having the same index).

8. Store the position coordinates (x_d, y_d) of the domain block and its calculated moment index value (I_d) in a temporary array (L) of records.
9. Sort the records of the array (L) in acceding order according to their moment based (I_d) index value.
10. Establish a set of pointers (p) refer to the start and end of each block of record hold same I_d value.
11. For each range block do the following:
    a. Calculate the average ($\bar{r}$)
    b. Quantize range average ($\bar{r}$) according to following equations

$$I_r = round(\tfrac{\bar{r}}{Q_r}) \quad \text{..... (21)}$$

$$\tilde{r} = Q_r I_r \quad \text{..... (22)}$$

Where

$$Q_r = \tfrac{255}{2^{br}-1} \quad \text{..... (23)}$$

Where

Q is the quantization step of the mean coefficients.

I_r is the quantization index of the mean coefficients.

b_r is the number of mean bits.

c. Calculate the moments $M_r(0,1)$, $M_r(1,0)$, $M_r(0,3)$, and $M_r(3,0)$.

d. Determine the block isometric index sym_indx_r that based on moment order-1 (table (1)).

e. Calculate the moment descriptors R_{1r} and R_{3r} using equations (18) and (19), respectively.

f. Calculate the block index (Ix_r) using equation similar to (20).

12. With help of a pointers set (p) and the temporary list of records (L); match only the domain blocks whose I_d values equal to I_r.
13. Sym_indx_r and sym_indx_d are passed through the isometric predictor, and then the predictor outputs the index of the required isometric transform (using Table (2)). Apply this transformation on the tested range block.
14. Calculate the scale *(s)* coefficient and (χ^2).
15. Compare the result (χ^2) of each matching instance with the minimum (χ^2) registered during the previous matching instance. If (χ^2) is smaller then put its value in minimum (χ^2) register (beside to the associated values of $(I_s, I_r, sym, x_d, y_d)$.
16. Check if $\chi^2_{min} < \varepsilon$ then the search across the domain blocks is stopped, and the registered domain block is considered as the best matched block and output the set $(I_r, I_s, sym, x_d, y_d)$ as best encountered IFS match, and go to step (11).
17. Otherwise, start the search for the domain blocks that have new Ix_d (belong to closest higher class) to get the best IFS match, if we haven't reach to an acceptable match instance try to match the domain blocks belong to other neighbour class and so on, until either the registered minimum error become less than ε (i.e, minimum allowable class) or all the domain bocks are tested. The new Ix_d values are computed as follows:

$$R_{1r}=R_{1r} \pm x \quad \text{..... (24)}$$

$$R_{3r}=R_{3r} \pm y \quad \text{..... (25)}$$

$$Ix_d = R_{1r} \times (N_m + 1) + R_{3r} \quad \text{..... ...(26)}$$

Where x and y values changes about the X, Y coordinates. The spiral search technique that based on city-block distance measure has been used to search for another neighbour class in four directions ((x, y), (x, -y), (-x, y), (-x, -y)) to reach the optimal one that registered minimum error.

18. Output set of IFS code (I_s,I_r , sym, x_d, y_d) for the tested range block.
19. Repeat steps (11) to (18) for all range blocks listed in the range pool.
20. Store all IFS mapping parameters as an array of record. The length of this array is equal to the number of range blocks in the range pool.

Decoding Process

The decoding process is summarized by the following steps:

1. Apply the dequantization process on the reconstructed i_s and i_r values to get the scale (s) and the mean ($\bar{r}$) values respectively for each range block as follow:

$$s = I_s \times Q_s \quad \text{..... (27)}$$

$$\bar{r} = I_r \times Q_r \quad \text{..... (28)}$$

2. Initialize the range pool by setting $\bar{r}_i$ values to its elements.
3. Generate the domain pool by down sampling (by 2) the range pool.
4. Apply IFS demapping to build approximates of the range block using the decoded IFS coefficients. This implies the application of the corresponding symmetry operation on the domain block and then mapping the resulted block (r_i') using equation (9).
5. Repeat steps 3, and 4 until reaching the attractor state (i.e. the difference between old and newly reconstructed range image is less than the predefined error or the number of the iteration exceeded maximum number of iterations.
6. The above steps (1-5) should be applied upon the three components Y, Cb, and Cr.
7. The two chromatic image Cb and Cr are upsampled to make all reconstructed bands have the same size (i.e. the image original size).
8. Convert (YCbCr) color components to RGB components using the inverse (YCbCr) transform.
9. Calculate the fidelity criteria (MSE, PSNR, CR, and BR) for the RGB reconstructed image.

Testing Results

The test results of both the developed system (i.e. using double descriptors) and the use of single descriptor method in this research are presented. The proposed system had been tested on Lena image (256x256, pixel 24bits) the size of range and domain blocks is taken 4×4 pixels;

Lena (24-bits RGB image).

As performance parameters, the PSNR, MSE, CR, BR and encoding time (ET) are computed. It is obvious that their values are significantly affected by the system parameters (i.e., N_m, b_s, b_r,, ε and S_{max}). The effects of each system parameter were investigated separately and the relevant test results have been listed as below:

Table (3) shows the effect of N_m on the performance of the proposed compression scheme (double moment descriptor). N_m is the maximum moment index, so the number of classes=$N_m \times(N_m+1)+N_m$. The results shows that the ET is decreased when N_m is increased. The good reduction in encoding time will occur with N_m=50 without cause a significant degradation in image quality.

Table (3)
The Effect of N_m on Performance of Using Double Moment Descriptor with Symmetry Predictor.

N_m	MSE	PSNR	ET(sec)
10	60.65	30.3	0.61
20	56.31	30.62	0.47
30	53.89	30.81	0.42
40	53.35	30.85	0.38
50	55.09	30.71	0.31
60	59.05	30.41	0.31

Table (4) shows the effect of N_m which is represent the number of classes, on the performance of the single moment descriptor compression scheme, the results shows that the ET is decreased when N_m is increased. The good reduction in encoding time will occur with N_m=500 without cause a significant degradation in image quality.

Table (4)
The Effect of N_m on Performance of Using Single Moment Descriptor with Symmetry Predictor.

N_m	MSE	PSNR	ET(sec)
100	42.27	31.86	2.11
200	45.98	31.5	1.33
300	47.83	31.33	1.05
400	49.2	31.21	0.84
500	51.11	31.04	0.72
600	53.35	30.85	0.67

Table (5) illustrates the effect of the maximum scale variation on the behaviour of the proposed compression performance, the results indicate that the effect on MSE, and PSNR is significant but the effect on ET is less significant. The value (S_{max}=4) was adopted as the best value.

Table (5)
The Effect of S_{max} on Performance of the Proposed Method.

S_{max}	MSE	PSNR	ET(sec)
1	79.06	29.15	0.39
2	55.59	30.68	0.39
3	55.11	30.71	0.36
4	55.09	30.71	0.34
5	72.52	29.52	0.34
6	74.9	29.38	0.36

The test results listed in Table (6) illustrates the effect of using the error threshold ε as stopping search condition on the compression performance. Both ET and PSNR are decreases when ε increases. The value of ε that was used in the proposed scheme is set to be 15 * block size^2.

Table (6)
The Effect of Error Throshold on the Performance of the Proposed Scheme.

ε	*MSE*	*PSNR*	*ET(sec)*
1	49.35	31.19	0.84
5	52.08	30.96	0.53
10	55.09	30.71	0.34
15	57.07	30.56	0.31
20	58.79	30.43	0.27
25	60.48	30.31	0.25

Table (7) shows the effectiveness of the quantization scale bits (i.e. bs) on the compression performance. It is clear that the quality of the of reconstructed image increased when the quantization bits were increased. When the scale bits become higher than 6, its variation became less effective. So the value 6 was adopted to be the best compromising value.

Table (7)
The Effect of Bs on the Proposed Fic.

Bs	*MSE*	*PSNR*	*ET(sec)*	*CR*	*BR*
2	114.94	27.52	0.56	9.14	2.62
3	70.75	29.63	0.42	8.82	2.71
4	61.9	30.21	0.39	8.53	2.81
5	56.36	30.62	0.36	8.25	2.9
6	55.09	30.71	0.34	7.99	3
7	54.65	30.75	0.34	7.75	3.09

Table (8) illustrates the effectiveness of the quantization mean bits (i.e. br) on the compression performance. The value 7 was adopted to be the best compromising value.

Table (8)
The Quantization Mean Bits br Effectiveness on the Proposed Method.

Br	*MSE*	*PSNR*	*ET(sec)*	*CR*	*BR*
4	207.42	24.96	0.44	8.82	2.71
5	91.69	28.5	0.34	8.53	2.81
6	62.23	30.09	0.34	8.25	2.9
7	55.09	30.71	0.31	7.99	3

Table (9) shows the difference between performing the proposed method with symmetry predictor and without symmetry predictor. The comparison depend on the effectiveness of the using same system parameters (i.e, b_s=6, b_r=7, ε =15 and S_{max}=4) on the performance parameters (PSNR, MSE, CR, BR and ET).

Table (9)
The Comparison between the Proposed System with Symmetry and Without Symmetry.

Parameter	*Proposed method without symmetry predictor*	*Proposed method with symmetry predictor*
MSE	71.64	57.07
PSNR	29.57	30.56
ET	0.45	0.31

Table (10) shows the comparison between the proposed method (the use of double moment descriptions) and the use of single moment descriptor method to speed up the FIC. The comparison depend on the effectiveness of the using same system parameters (i.e, b_s=6, b_r=7, ε =15 and S_{max}=4) on the performance parameters (PSNR, MSE, CR, BR and ET).

Table (10)
The Comparison between the Double and Single Moment Descriptor.

Parameter	*Double Descriptor with N_m=50*	*Single Descriptor with N_m=500*
MSE	57.07	52.95
PSNR	30.56	30.89
ET	0.31	0.61
CR	7.99	7.99
BR	3	3

Conclusions

The conclusions have been drawn from this work are listed below:

1. The results shows that the using of double moment descriptors with symmetry predictor decreasing the encoding time approximately 50% from the ET of using single moment descriptor without making significant degradation in image quality.
2. The test results indicate that the usage of symmetry predictor with the double block descriptors causes a decrease in both encoding time and MSE (increase the PSNR). This is because the proposed

indexing method gave us more precisely classification to range and domain blocks.

3. The proposed algorithm could be modified to imply quadtree partitioning scheme to increase the compression ratio.

References

[1] M. F. Barnsley and L. Hurd, "Fractal Image Compression". A. K. Peters, Wellesley, M/a, 1993.

[2] M. F. Barnsley, "Fractals Everywhere". Academic Press, 1993.

[3] A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation". IEEE Transaction on Image Processing. 1992, vol.1, no.1 pp.18-30, 992.

[4] Y. Fisher, Fractal Image Compression, SIGGRAPH Course Notes, 1992

[5] N. T. Thao, K. Asia, and Vetterli, "Set Theoretic Compression with an Application to Image Coding". Proceedings IEEE International Conference Image Processing, vol. II, pp. 336-340, 1994.

[6] D. J. Hebert and E. Soundararajan, Fast Fractal Image Compression with Triangulation Wavelets". Proceeding SPIE Conference on Wavelet Application in Signal and Image Processing, pp.67-74, 1998.

[7] L. E. George, E. A. Al-Hilo,*" Speeding-Up Fractal Colored Image Compression Using Moments Features ", December 2008*.

[8] L. E. George, E. A Al-Hilo," *Fractal color image compression By Adaptive Zero-Mean Method*", IEEE, ISBN: 978-1-4244-5483-2, (ICCTD 2009) Malaysia, Nov. 13-15, Vol.1, page (525-530), 2009.

[9] L. Ning, "*Fractal Imaging*", book, Academic Press, 1997.

[10] R. Gonzalez, R. Wood, "Digital Image Processing", Person Education International, Prentice Hall, Inc. 2nd Edition 2002.

[11] S. J. Sangwine, R. Horne, "*The Color Image processing Handbook*", Champan &Hall, 1998

[12] L. E. George, "*IFS Coding for Zero-Mean Image Blocks*", Iraqi Journal of Science, vol.47, no.1, 2006.

[13] C. Frigaard, "Fast Fractal 2D/3D Image Compression", Report, Institute of Electronic Systems, an Alborg University, Laboratory of Image Analysis, 1995.

الخلاصة

في هذا البحث تم تطوير نظام الضغط الكسوري المعتمد على تجزئة نظام الدالة التكرارية من خلال تقليص وقت الضغط. الطريقة المستحدثة اعتمدت على استخدام متنبيء التحويل التناظري و على استخدام واصف للمقاطع يعتمد على العزوم من الدرجة الاولى و الثالثة، فقد استخدم واصف المقاطع لعمل فهرسة البلوكات للمنطلق، وبالنتيجة ليختزل معادلات البحث عن افضل بلوك منطلق شبيه لكل بلوك من بلوكات المدى. تم اعتماد فهرسة البلوكات على استخدام واصف مزدوج ينتج عن استخدام العزوم من الدرجة الاولى والثالثة مع بعض ليعطينا فهرس واحد بلا من استخدام عزم واحد. ان نتائج فحص هذين الهيكلين اشارت الى حصول اختزال بالوقت اي نقصان في وقت الضغط ليصل الى ٠,٢٥ ثانية بدون ان يسبب تغير ملحوظ في نوعية الصورة مع اعطائنا نسبة ضغط تصل الى ٧,٩٩.