DOI: 10.24996/ijs.2021.62.12.39

ISSN: 0067-2904

# Fast Fractal Technique using Modified Moment Features on Domain Blocks

**Alaa Ali Hussein[1]*, Atheer Yousif Oudah[2]**

*[1] Department of Research and Development, Ministry of Higher Education and Scientific Research, Baghdad, Iraq*

*[2]Computer Science and Thi-Qar University/ Faculty of Education for Pure Science, Thi-Qar, Iraq*

Received: 30/**12/**2020 Accepted: 26/6/2021

**Abstract**

In this research, a new technique is suggested to reduce the long time required by the encoding process by using modified moment features on domain blocks. The modified moment features were used in accelerating the matching step of the Iterated Function System (IFS). The main disadvantage facing the fractal image compression (FIC) method is the over-long encoding time needed for checking all domain blocks and choosing the least error to get the best matched domain for each block of ranges. In this paper, we develop a method that can reduce the encoding time of FIC by reducing the size of the domain pool based on the moment features of domain blocks, followed by a comparison with threshold (the selected threshold based on experience is 0.0001). The experiment was conducted on three images with size of 512x512 pixel, resolution of 8 bits/pixel, and different block size (4x4, 8x8 and, 16x16 pixels). The resulted encoding time (ET) values achieved by the proposed method were 41.53, 39.06, and 38.16 sec, respectively, for boat , butterfly, and house images of block size 4x4 pixel. These values were compared with those obtained by the traditional algorithm for the same images with the same block size, which were 1073.85, 1102.66, and 1084.92 sec, respectively. The results imply that the proposed algorithm could remarkably reduce the ET of the images in comparison with the traditional algorithm.

**Keywords:** Partitioned Iterated Function System , Domain Blocks, Encoding , Moments Features

# تسريع التقنية الكسورية باستخدام مميزات العزم المعدل على كتلة المجال

**علاء علي حسين[1]* , اثير يوسف عودة[2]**

[1]دائرة البحث والتطوير، وزارة التعليم العالي والبحث العلمي، بغداد، العراق

[2]علوم الحاسبات، كلية التربية للعلوم الصرفة، ذي قار، العراق

**الخلاصة**

في هذا البحث، تم اقتراح تقنية جديدة لتقليل عملية التشفير لفترة طويلة باستخدام ميزات العزم المعدلة على كتل المجال. تسريع خطوة مطابقة نظام الوظائف المتكررة (IFS) باستخدام ميزات العزم المعدل. العيب

* Email ali.alaa597@gmail.com

الرئيسي في مواجهة تقنية ضغط الصور الكسورية (FIC) هو وقت التشفير الطويل جدًا نظرًا للعملية التي يتم إجراؤها من خلال فحص جميع كتل المجال واختيار أقل خطأ للحصول على أفضل مجال مطابق لكل كتلة من النطاقات. في هذا البحث ، قمنا بتطوير طريقة يمكن أن تقلل من وقت الترميز لهذه التقنية عن طريق تقليل حجم تجمع المجال بناءً على ميزات العزوم الخاصة بكتل المجال ثم المقارنة مع العتبة (العتبة المحددة من خلال التجربة 0.0001). ثلاث صور (512 × 512) دقة 8 بت / بكسل مع كتل مختلفة الحجم (4 × 4 ، 8 × 8 , 16 × 16) بكسل ، وقت الترميز الناتج (ET) بالطريقة المقترحة لحجم الكتلة هو (4 × 4 بكسل) للقارب والفراشة والمنزل كانت الصور 41.53 ثانية و 39.06 ثانية و 38.16 ثانية على التوالي. عند المقارنة مع وقت التشفير (ET) ، كانت قيم الخوارزمية التقليدية لحجم الكتلة (4 × 4 بكسل) من نفس الصور 1073.85 ثانية و 1102.66 ثانية و 1084.92 على التوالي. هذا يعني أن الخوارزمية المقترحة للصور ذات قيم وقت التشفير (ET) أفضل من قيم وقت التشفير (ET) للخوارزمية التقليدية.

## 1. Introduction

A new image compression technique, called fractal compression, has become rapidly popular. Nowadays it is being used in many implementations, including texture segmentation, image compression, image signatures, feature extraction, and image watermarking [1]. Fractal image compression (FIC) is a method of lossy compression, attempting to construct an approximation of the original image. The major task of this method is to examine similarities between large and small parts of images [2]. FIC is utilized in various image processing applications, such as image retrieval, feature extraction, image signature, multi-resolution medical image processing, and texture segmentation [3]. It has important advantages, such as high compression ratio and fast image reconstruction. Also, its multi-resolution property is one of the advantages achieved by decoding the image to get lower or higher resolutions than the original image. In addition, it is possible to zoom-in on the different parts of the image [4]. The major problem facing FIC is that the similarity matching process between the range blocks and domain blocks is complex. This problem could be solved through an effective technique that reduces the size of the domain pool based on the moment features of domain blocks, followed by a comparison with a threshold value before the matching process, so as to make the encoding process faster [5].

## 2. Traditional PIFS Encoder

The basic idea of Partitioned Iterated Function System (PIFS) is dividing the image into non-overlapping range blocks by quadtree, fixed with Horizontal Vertical (HV) partitioning [6]. The domain pool creation module is responsible for the creation of a domain array ($H_d$x $W_d$) which is quarter the size of the original image order (H xW),(i.e.,$H_d$= H/2 and $W_d$= W/2). The data of the domain array is taken from the range array; there are many methods to choose the data from the range array to fill the domain array. For each four neighboring pixels in the range array, only one value of these pixels is stored in the domain array. In this work, the average of each four neighboring pixels in the range is applied to filling in the identical position in the area [7].

After generating the range and domain library (or domain pool), the matching process is applied for each range block with all domain blocks (D) recorded in the domain library (or domain pool). In order to achieve a minimum distortion error R, there should be the best match between the domain block and range block [8].

$$R = \frac{1}{n}\left[\sum_{i=1}^{n} b_i^2 + s\left(s\sum_{i=1}^{n} a_i^2 - 2\sum_{i=1}^{n} a_i b_i + 2o\sum_{i=1}^{n} a_i\right) + o\left(no - 2\sum_{i=1}^{n} b_i\right)\right] \quad (1)$$

The IFS coefficients of equation (1), i.e. the scale "S' and the substitute "O", are determined by using the following equations:

$$s = \frac{[n.(\sum_{i=1}^{n} a_i b_i) - (\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} b_i)]}{[n \sum a_i^2 - (\sum_{i=1}^{n} a_i)^2]} \quad (2)$$

$$o = \frac{\left[\sum_{i=1}^{n} b_i - s \cdot \sum_{i=1}^{n} a_i\right]}{n} \quad (3)$$

whe
$a_i$ : is the pixel value of the domain block.
$b_i$ : is the pixel value of the range block.
n: is the block size.

If $n \sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} a_i)^2 = 0$ then S = 0 and

$$o = \frac{1}{n} \sum_{i=1}^{n} b_i$$

There is a simple formula for R, which is: rms = $\sqrt{R}$ [9].

The quantization of the IFS coefficients is performed by assigning a number of bits for substitute and scale to store their quantization indices. By using equations (4), (5), and (6), the quantized substitute and scale values are computed [10, 11].

$$Q_s = \begin{cases} \frac{S_{max}}{2^{b_s-1}} & if\ S_{\max} = -S_{min} \\ \frac{S_{\max} - S_{\min}}{2^{b_s-1}} & if\ S_{\max} \neq -S_{min} \end{cases} \quad (4)$$

$$i_s = \text{round}\left(\frac{S}{Q_s}\right) \quad (5)$$

$$S_q = i_s Q_s \quad (6)$$

where:
$S_{max}$ : is the highest allowable value of scale coefficients.
$S_{min}$ : is the lowest allowable value of scale coefficients.
$b_s$ : is the number of allocated bits.
$Q_s$ : is the quantization stage of the scale coefficients.
$I_s$ : is the quantization index of the scale coefficients.
$S_q$ : is the quantized value of the scale coefficients.

$$Q_o = \begin{cases} \frac{O_{max}}{2^{b_o-1}} & if\ O_{\max} = -O_{min} \\ \frac{O_{\max} - O_{\min}}{2^{b_o-1}} & if\ O_{\max} \neq -O_{min} \end{cases} \quad (7)$$

$$I_o = \text{round}\left(\frac{O}{Q_o}\right) \quad (8)$$

$$O_q = i_o Q_o \quad (9)$$

where:
$O_{max}$ : is the highest allowable value of substitute coefficients.
$O_{min}$ : is the lowest allowable value of substitute coefficients.
$b_o$ : is the number of allocated bits .

$Q_o$ : is the quantization stage of the substitute coefficients.
$I_o$ : is the quantization index of the substitute coefficients.
Oq : is the quantization value of the substitute coefficients.
The quantized values of scale (S) and substitute (O) parameters should be used to calculate the sum of square error R using equation (1). The value of the error ® is compared with the minimum recorded error ($R_{min}$). The IFS parameters are calculated and the sum of the error, for any matching case between the range and each domain of the block, is recorded in the domain pool [12] .

If $R< (R_{min})$ then

$S_{opt} = i_s$ ;$O_{opt} = i_o$ ; $R_{min}= R$

End if [6].

In the decoding stage, the approximations could be performed several times, starting with any random image until reaching the fixed point [6].

**2. Usage of Moments in the Modified PIFS Encoder**

The range and domain pools are generated first, where f(x,y) represents image block and the around its center point (xc,yc). The moment of order (p+q) of block (f) is defined as [7]:

$$M(p,q) = \sum_y \sum_x (x - x_c\ )^p\ (y - y_c)^q [f(x, y) - f^-(x, y)] \qquad (10)$$

When this definition is applied to determine the first order moments of the domain and range blocks, the following expressions are obtained:

$$M_d(1,0) = \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} (x - k_c)(d_{ij} - d^-) \qquad (11)$$

$$M_d(0,1) = \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} (y - k_c)(d_{ij} - d^-) \qquad (12)$$

$$k_c = \frac{k-1}{2} \qquad (13)$$

Now, the moments ratio factor (*R*) is:

$$R=\frac{m^2(0,1) - m^2(1,0)}{m^2(0,1)+m^2(1,0)} \qquad (14)$$

So, the moment ratio factors value for domain is:

$$R_d= \frac{m_d^2(0,1) - m_d^2(1,0)}{m_d^2(0,1)+m_d^2(1,0)} \qquad (15)$$

The moment value of domain blocks is compared with a threshold value. If the moment value is lower than the threshold, then we compute the matching between the domain and range blocks. The processes involved in this modified moment can be summarized by the following steps:

Step1: Load the image.
Step2: Choose the threshold.
Step4: The image is subdivided into fixed-sized blocks with non-overlap (R1… Rn-1 ).
Step5: Generate the domain image from the original image by the averaging method.
Step 6: Build new domain blocks (D1...$D_{n-1}$ ).
Step7: Do the mapping process by

- compute the moment by using equation (10) for each new domain block
- compute the scale (S) and substitute (O) coefficients according
- Compute the s and o values

Step8: Compare the moment of the domain block with threshold.

a) If the moment value of the domain block is lower than the threshold
b) then $D_{n-1}$ is the best matching:

- Compute RMS using equation (1). If RMS is less than €, then store the IFS code; else, go to the next domain block.

## 4. Experimental Results

The proposed algorithm was tested by using gray scale images of boat , butterfly, and house (512x512 pixels). The execution of the proposed algorithm was carried out on a laptop Intel® Core i5-2410; 4 GB RAM 500 GB Hard-Disk with Windows 10, using the modified moment features through the MATLAB environment facilities. The experiment was implemented with the technique of quad tree partitioning, allowing up to three block sizes (4x4, 8x8, and 16x16 pixels) to be examined. Table (1) lists the results obtained by using the traditional fractal images compression method. The results tables (2) , (3) and (4) show the encoding time with the values of PSNR and CR. The results of the proposed algorithm for boat , butterfly, and house images are presented in figures (2b) , (4b), and (6b) , respectively. Figures (1), (3), and (5) show the results of the encoding time for the tree values of block size for both the moments and the traditional algorithms.

**Table 1-**Test results of PSNR, ET, and CR values for images (512x512 pixels) processed with the traditional algorithm;
block size ( 4x4, 8x8 and 16x16 pixels).

| Image | BLOCKSIZE | PSNR | ET | CR |
|---|---|---|---|---|
| boat | 4x4 | 37.14 | 1073.85 | 6.02 |
| | 8x8 | 36.06 | 1006.14 | 11.17 |
| | 16x16 | 31.57 | 914.82 | 19.08 |
| butterfly | 4x4 | 35.63 | 1102.66 | 6.97 |
| | 8x8 | 33.94 | 959.08 | 13.85 |
| | 16x16 | 29.73 | 890.29 | 21.93 |
| house | 4x4 | 37.28 | 1084.92 | 4.94 |
| | 8x8 | 31.96 | 947.68 | 11.89 |
| | 16x16 | 26.51 | 878.39 | 21.06 |

**Table 2-** Test results of the processing of the boat image (512x512 pixels) using the proposed moments method; block size ( 4x4, 8x8 and 16x16 pixels), threshold ( the selected threshold value based on experience is 0.0001) .

| boat (512x512) pixels | | | | |
|---|---|---|---|---|
| THRESHOLD | BLOCKSIZE | PSNR | ET | CR |
| 0.0001 | 4x4 | 31.95 | 41.53 | 5.14 |
| 0.0001 | 8x8 | 28.36 | 20.74 | 12.42 |
| 0.0001 | 16x16 | 23.17 | 15.06 | 20.37 |

.

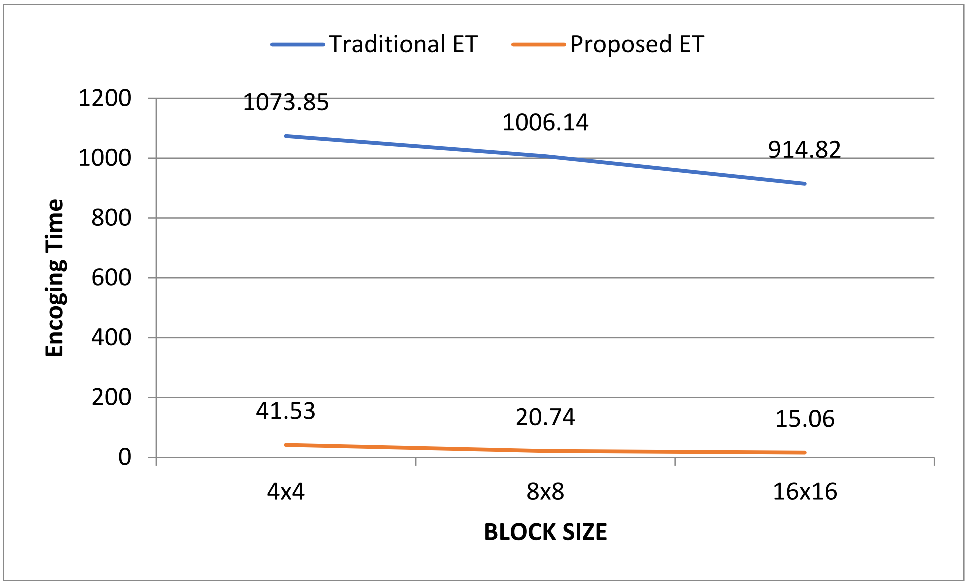


**Figure 1**-Comparison of encoding times (ET) between the proposed and traditional algorithms


**Figure 2**-(a) original boat image (512x512 pixels), (b) boat image (512x512 pixels) processed using the moments algorithm; PSNR= 31.9514, ET= 41.53811, Block size= 4x4 , CR=5.14.

**Table 2**-Test results of butterfly image (512x512 pixels) with the proposed moments method; block size ( 4x4, 8x8 and 16x16 pixels) , threshold (the selected threshold value based on experience is 0.0001) .

| butterfly(512x512) pixels | | | | |
|---|---|---|---|---|
| THRESHOLD | BLOCKSIZE | PSNR | ET | CR |
| 0.0001 | 4x4 | 29.27 | 39.06 | 6.27 |
| 0.0001 | 8x8 | 24.42 | 18.13 | 13.39 |
| 0.0001 | 16x16 | 21.63 | 13.94 | 20.94 |

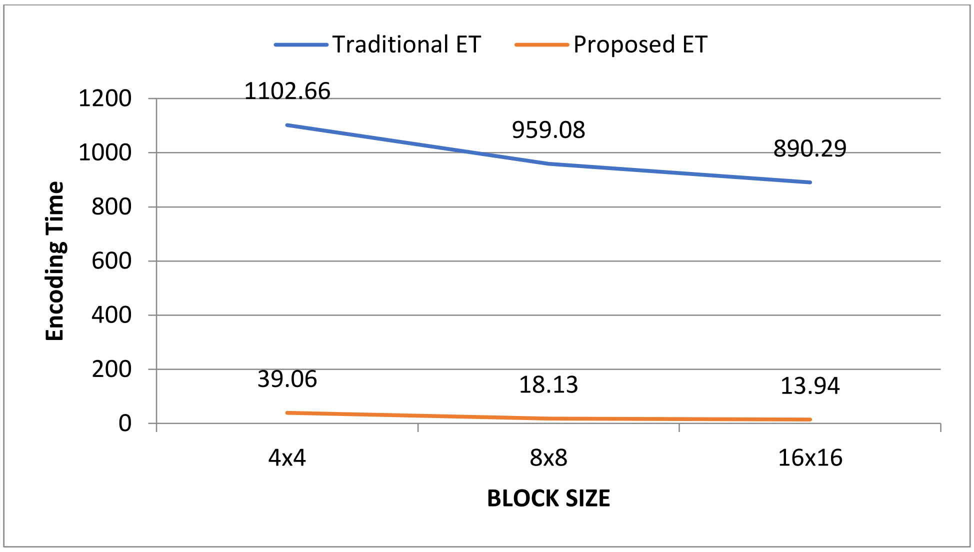


**Figure 3-** Comparison of encoding time (ET) values between the proposed and traditional algorithms.


**Figure 4** -(a) original butterfly image (512x512 pixels), (b) butterfly image (512x512 pixels) processed using the moments algorithm; PSNR=29.2702, ET= 39.062811, Block size= 4x4, CR=6.27.

**Table 3-**Test results of the house image (512x512 pixels) with the proposed moments method ; block size (4x4, 8x8 and 16x16 pixels) , threshold( the selected threshold value based on experience is 0.0001) .

| House (512x512) pixels | | | | |
|---|---|---|---|---|
| THRESHOLD | BLOCKSIZE | PSNR | ET | CR |
| 0.0001 | 4x4 | 30.52 | 38.16 | 5.82 |
| 0.0001 | 8x8 | 24.34 | 17.81 | 11.96 |
| 0.0001 | 16x16 | 20.76 | 13.24 | 20.47 |

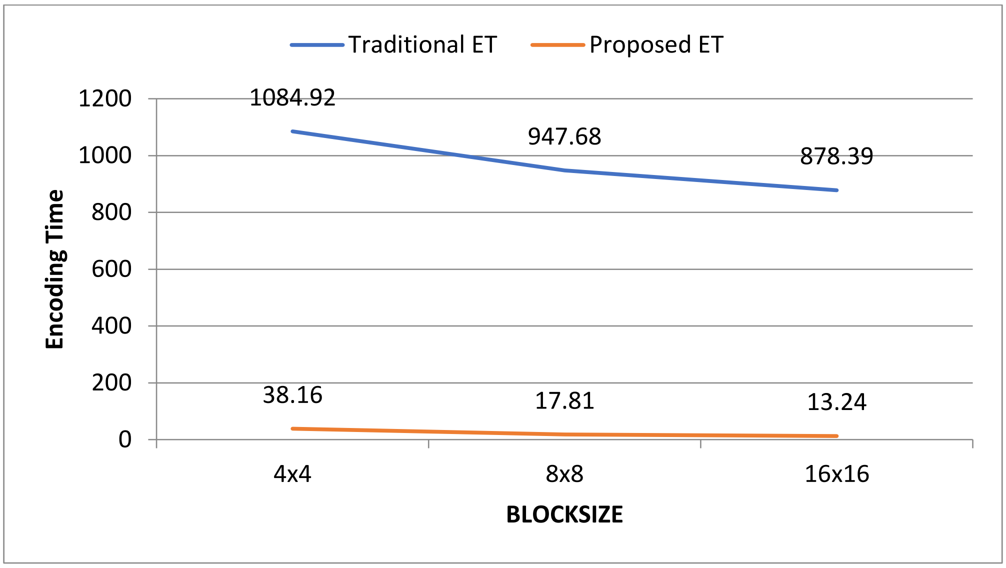


**Figure 5-**Comparison of encoding times (ET) between proposed and traditional algorithms.


**Figure 6-**(a) original house image (512x512 pixels) , (b) house image (512x512 pixels) processed using the moment algorithm; PSNR=30.5218, ET= 38.719046, Block size= 4x4, CR=6.27.

## 5. Conclusions

Our experimental results were based on the comparison of the traditional fractal image compression algorithm and the proposed method based on the moment technique. We observed that the highest ET value of boat image (block size 4x4) using the proposed method was 41.53sec, while that obtained using the traditional algorithm (was 1073.85sec. This means that the encoding time of the traditional algorithm was decreased about 1032.32 sec when the proposed algorithm was applied. This implies that we could achieve the aim of overcoming the long encoding time problem. In addition, our results showed that when the size of the block increased, the ET and PSNR values decreased and C.R. increased.

## References

**[1]** Zainab J. Ahmed1 , Loay E. George and Zinah S. Abduljabbar. "Fractal Image Compression Using Block Indexing Technique: A Review" *Iraqi Journal of Science*, vol. 61, no. 7, pp: 1798-1810, 2020.

**[2]** Shaimaa S. AL-Bundi and Mustafa S. Abd. A Review on Fractal Image Compression Using Optimization Techniques. Journal of Al-Qadisiyah for Computer Science and Mathematics vol.12, no,1. pp 38–48, 2020.

**[3]** Salarian.M, Nadernejad.E, Naim. H. A new modified fast fractal image compression Algorithm,The Imaging Science Journal vol 61, pp. 219-231, 2013.

**[4]** Li Hsu. S., Chang.Y., Jeng. J.," A Study of Fractal Image Compression with Coefficient Quantization", Department of Information Engineering I-Shou University, Ta-Hsu Hsiang, Kaohsiung County, The 24th Workshop on Combinatorial Mathematics and Computation Theory, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.378.9750&rep=rep1&type=pdf

**[5]** George. L. E, Minas. N. A. Speeding Up Fractal Image Compression Using DCT Descriptors. Journal of Information and Computing Science, vol.6. no.4, pp 287-294, 2011.

**[6]** Al-Hilo. E.,"Speeding-up Fractal Colored Image Compression Using Moments Features." Dec. 2008. DOI**:** 10.1109/DICTA.2008.18.

**[7]** Taha M.H. Partitioning Development for Fractal Image Compression. M.Sc. Thesis, Science College, AL-Mustansiriyah University. 2005.

**[8]** Hussein A.A. 2003. Fractal Image Compression with Fasting Approaches. M.Sc. Thesis, College of Science, Saddam University.

**[9]** Fisher Y.1994. Fratal Image Compression :Theory and Application. SpringierVerlage, New York.

**[10]** AL-Ali. Hazeem. B.Taher.2013. Speeding- Up Audio Fractal Compression Using Double Moments Descriptor",Ph.D. Thesis, College of Science, Baghdad University.

**[11]** George\L.. "IFS Coding for Zero-Mean Image Blocks." *Iraqi Journal of Science* ,vol.47,no.1,pp. 190-194, 2006.

**[12]** Haider A. and Hazeem. B. "Speeding- Up Fractal Image Compression Using Entropy Technique" *International Journal of Computer Science and Mobile Computing*, vol.5, no.4, pp. 518-524, 2016.