Study the Performance of Fractal Gray Image Compression Using Different Images Size

Eman A. Al-Hilo
Kufa University/ College of Education for girls/ Kufa/Iraq
emanalhilo@yahoo.com

Kawther H. Al-khafaji
Kufa University /College of Education for girls/ Kufa/Iraq
Kutherh.kafajy@uokufa.equ.iq

Abstract

In this paper we investigate the effect of image size on the compression parameters of the fractal image compression technique(FIC) proposed by Jacquin This research is tested on 8 bits/pixel gray images, three different size of gray image have been used (100 x 100, 150 x 150 ,256 x256). The results show that the PSNR, Bit Rate and Encoding time are increase with the increase of image size, but the CR is decreases with the increase of image size, The quality of the reconstructed image is pleasurable either we use any size of images .

Key Words: Image Compression, , fractal image compression.

دراسة أداء الضغط ألكسوري للصورة الرمادية باستخدام أحجام صور مختلفة

إيمان عبد المالك الحلو
جامعة الكوفة / كلية التربة للبنات

كوثر حسن صاحب الخفاجي
جامعة الكوفة / كلية التربة للبنات

الخلاصة

تم في هذا البحث التحقق من تأثير حجم الصورة على عوامل الضغط لتقنية ضغط الصور ألكسوري FIC والمقترحة من قبل جاكوين , الاختبارات التي أجريت في هذا البحث تم فيها استخدام مجموعة من الصور الرمادية (8 bit) بثلاثة إحجام مختلفة وهي (100x100,150x150,256x256) أثبتت النتائج العملية إن إي زيادة في حجم الصورة يقود إلى زيادة في قيم PSNR و BitRate وكذلك في زمن التشفير ET , بينما تؤدي إلى تناقص في قيم نسبة الضغط CR , ولكن على العموم فان جودة الصورة المسترجعة كانت تقريبا جيدة في كل حالة.

كلمات مفتاحيه:- ضغط الصور، ، ضغط الصور ألكسوري.

1. Introduction

The idea of fractal image compression (FIC) was originally introduced by Barnsley and the first practical FIC scheme was realized by Jacquin in 1992. The underlying premise is based on the partitioned iteration function system (PIFS) which utilized the self-similarity property in the image to achieve the purpose of compression. The encoding process of the fractal image compression is time-consuming. The reason is that most of the encoding time is spent on a large amount of computations of similarity measure. [LICH12]

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. Therefore the 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. [Saa10]

2. Basic Fractal Image coding

As the compression method, the implemented fractal compression Scheme consists of two major units ,the first unit is the encoding unit and the second one is the decoding unit. Each of these two units consists of many modules. [Hilo07]

The main theory of fractal image coding is based on iterated function system, attractor theorem and Collage theorem. Fractal Image coding makes good use of Image self-similarity in space by ablating image geometric redundant. Fractal coding process is quite complicated but decoding process is very simple, which makes use of potentials in high compression ratio.[ChRa09]

3. Encoding Unit

This unit consists of three modules which are all together responsible for reducing the data size of the desired gray image and generate the PIFS as a compressed stream of data to represent the image. As shown in Figure (1)

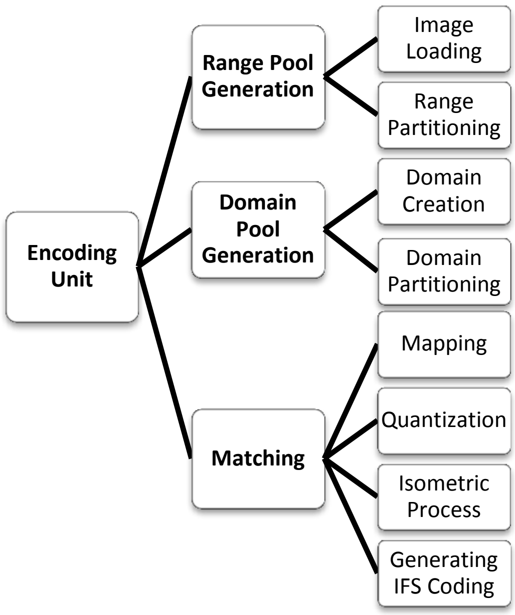


Fig (1) Encoding Unit

For a range block with pixel values ($r_o, r_1, \ldots, r_{n-1}$), and the domain block ($d_o, d_1, \ldots, d_{n-1}$) the contractive affine approximation is[GeHi11]:

$$\grave{r}_i \approx s d_i + o, \ldots\ldots\ldots\ldots (1)$$

Where, $\grave{r}_i$ is the optimally approximated i[tch] pixel value in the range block. d_i is the corresponding pixel value in the domain block. The symbols s,o represent the scaling and offset coefficients, respectively.

These parameters (s) and (o) are determined by applying the least sum ϵ^2 of square errors between ($\grave{r}_i$)and (r_i) according to following equation [Fish95]:

$$\epsilon^2 = \sum_{i=0}^{n-1} (\grave{r}_i - r_i)^2, \ldots\ldots\ldots\ldots (2)$$

The minimum of ϵ^2 occurs when:

$$\frac{\partial \epsilon^2}{\partial s} = 0 \quad \text{and} \quad \frac{\partial \epsilon^2}{\partial o} = 0 \ldots\ldots\ldots\ldots (3)$$

The straightforward manipulation of the above equation leads to:

$$s=\frac{n\sum_{i=0}^{n-1} r_i d_i-\sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} d_i}{n\sum_{i=0}^{n-1} {d_i}^2-\left(\sum_{i=0}^{n-1} d_i\right)^2}\ ,\ldots\ldots\ldots\ldots\ldots(4)$$

$$o=\frac{\sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} {d_i}^2-\sum_{i=0}^{n-1} r_i d_i \sum_{i=0}^{n-1} d_i}{n\sum_{i=0}^{n-1} {d_i}^2-\left(\sum_{i=0}^{n-1} d_i\right)^2}\ ,\ldots\ldots\ldots\ldots(5)$$

$$\epsilon^2=\frac{1}{n}\left[\sum_{i=0}^{n-1} r_i^2+s\left(s\sum_{i=0}^{n-1} d_i^2-2\sum_{i=0}^{n-1} r_i d_i+2o\sum_{i=0}^{n-1} d_i\right)+o\left(no-2\sum_{i=0}^{n-1} r_i\right)\right],\ldots\ldots\ldots(6)$$

The implementation encoding method could be summarized by the following steps:

A. Load BMP image and put it in (2D arrays).
B. Establish the range image (array).
C. Down sample the range image to produce the domain array.
D. Great range and domain pool by partitioning:

(1) The range array must be partitioned into non-overlapping fixed blocks, to generate the range blocks ($r_1, \ldots, r_n$).


Original image Range Pool

Fig (2) Range partitioning

(2) The domain must be partitioned into overlapping blocks, using specific step size, to generate the domain blocks ($d_1, \ldots, d_n$). They should have the same size of range blocks.

E. Searching: For each range block do the following:

(1) Pick up a domain block from the domain pool.
(2) Perform one of the isometric mappings.
(3) Calculate the scale (s) and offset (o) coefficient using equations (4,5).
(4) Apply the following condition to bound the value of (s) and offset (o) coefficient:

If $s < s_{min}$ *then* $s=s_{min}$
Else if $s > s_{max}$ *then* $s=s_{max}$
If $o < o_{min}$ *then* $o=o_{min}$
Else if $o > o_{max}$ *then* $o=o_{max}$

(5) Quantize the value (s) and offset (o) using equations refer in [9].
(6) Compute the approximation error (ϵ_{min}^2) using equation (6).
(7) After the computation of IFS code and the sum of error (ϵ^2) of the matching between the range and the tested domain block , the (ϵ^2) is compared with registered minimum error (ϵ_{min}^2); such that:

If $\epsilon^2 < (\epsilon_{min}^2)$ *then*
$s_{Opt}=i_s;\quad o_{Opt}=i_o;\quad \epsilon_{min}^2=\epsilon^2$
PosI=domain block index
Iso=isometric index
End if

(8) If $\epsilon_{min}^2 < \varepsilon$ then the search across the domain blocks is stopped, and the registered domain block is considered as the best matched block.
(9) Repeat steps (4) to (10) for all isometric states of the tested domain block.
(10) Repeat steps (3) to (11) for all the domain blocks listed in the domain pool.
(11) The output is the set of IFS parameters $(i.e., i_s, i_o, posI, Iso)$ which should be registered as a set of fractal coding parameters for the tested range block.
(12) Repeat steps (1) to (12) for all range blocks listed in the range pool
(13) 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.

4. Decoding Unit

This unit consists of two modules, as shown in Figure (2), it starts with loading the IFS code and ends with the attractor as output.

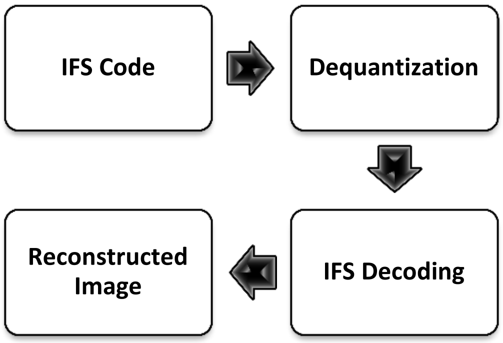


Fig (3) Decoding Unit

The decoding process can be summarized by the following steps:

1. Generate arbitrary domain pool, the domain pool could be initialized as a blank image or a piece of image extracted from any available image.
2. The values of the indices of (i_s) and (i_o) for each range block should be mapped to reconstruct the quantized values of the scale (s_q) and offset (o_q) coefficients.
3. Choose the number of possible iterations, and the threshold value of the mean square error (TMSE). At each iteration do the following steps:
   a. For each range block determine the coordinates (x_d,y_d), of the best matched domain, from the IFS parameters (*posI*), in order to extract the domain block (*d*) from the arbitrary domain image.
   b. For each range block, its approximation r_i' is obtained by multiplying the corresponding best matched domain block (d) by the scale value (s_q) and adding to the result the offset value (o_q), according to equation (1).
   c. The generated r_i' block is transformed (rotated, reflected, or both) according to its corresponding IFS isometric parameter value (*Iso*).
   d. Put the generated r_i' block in its position in the decoded image array (i.e., range image).
   e. Check whether there is another range block, if yes then repeat steps (b,c,d)
   f. Down sample the reconstructed image (range pool) in order to produce the domain pool using the averaging sampling.
   g. Calculate the mean square error MSE between the reconstructed range and the previous reconstructed range image. If the MSE is greater than TMSE value then the iteration continues and the above steps (a-f) should repeated; this iteration is continued till reaching the attractor state (i.e., the newly reconstructed range image is very similar to the previous reconstructed image). Otherwise the iteration continues till reaching the predefined maximum number of iterations.

5. Tests Results

The proposed system was established using Visual Basic *(Ver.6.0)* and tested on *Acer* laptop with (*500GHZ*, *RMA 2MB*).

The proposed system had been tested on three gray image *(8 bits)* with different size*(256x256,150x150,100x100)Pixel*,The value of the parameters MaxOffset and MinOffset were fixed in all these tests at *(255)* and *(-256)* respectively but the other coding parameters were taken as: BlockSize=4x4, StepSize=2, ScaleBits = 6 , OffsetBits=6, MinScale=-1.5 , MaxScale=3, ε =0.4, TMSE=0.05.

Table (1) illustrates the effect of image size on encoding time, PSNR, and CR.

The results show that PSNR and ET increase with the increase of image size but the CR is decreases with the increase of image size .

Table(1) Effect of image size on the compression performance

Image	Image Size	Domain Size	ET (sec)	PSNR (dB)	CR	Bit Rate	CF	MAD	MSE
Mandrill	100 x 100	50 x 50	1.42	28.93	5.27	4.55	0.19	6.68	83.21
	150 x 150	75 x 75	6.83	21.42	5.17	4.64	0.19	9.83	468.98
	256 x 256	128 x 128	61.93	28.07	4.74	5.06	0.21	7.07	101.49
Lena	100 x 100	50 x 50	1.42	29.77	5.27	4.55	0.19	5.07	68.50
	150 x 150	75 x 75	6.81	21.07	5.17	4.64	0.19	6.98	508.16
	256 x 256	128 x 128	59.86	34.75	4.74	5.06	0.21	2.53	21.79
Couple	100 x 100	50 x 50	1.42	29.34	5.27	4.55	0.19	5.75	75.65
	150 x 150	75 x 75	6.85	19.73	5.17	4.64	0.19	8.72	692.50
	256 x 256	128 x 128	61.81	32.46	4.74	5.06	0.21	3.75	36.88

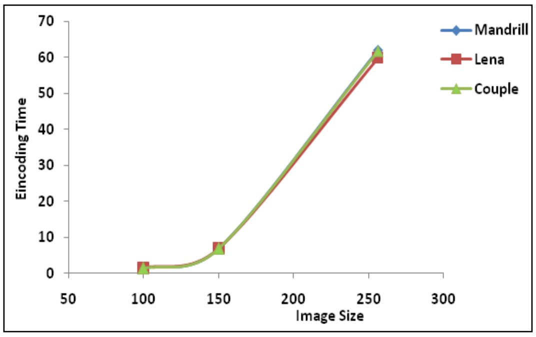


Fig (4) Effect of image size on ET

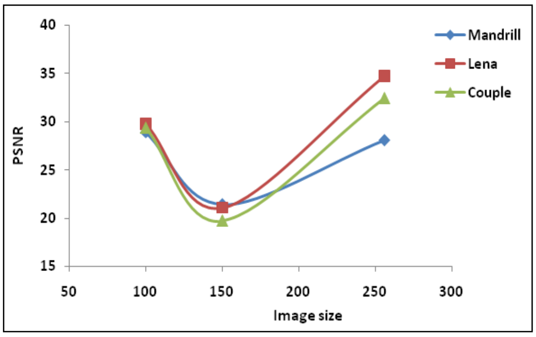


Fig (5) Effect of image size on PSNR

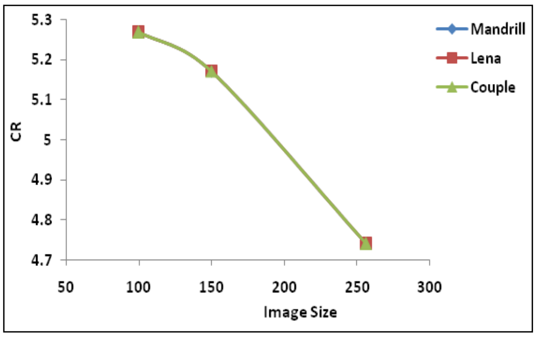


Fig (6) Effect of image size on CR

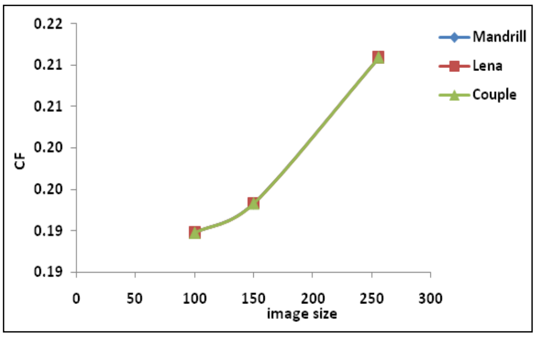


Fig (8) Effect of image size on CF

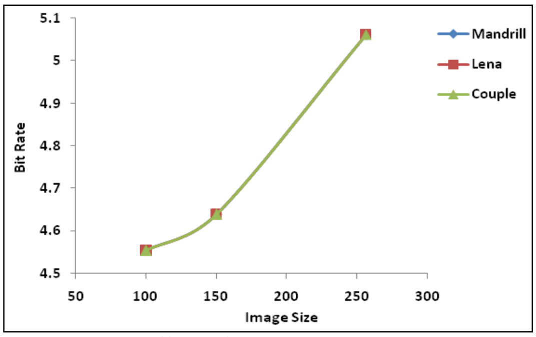


Fig (7) Effect of image size on Bit Rate

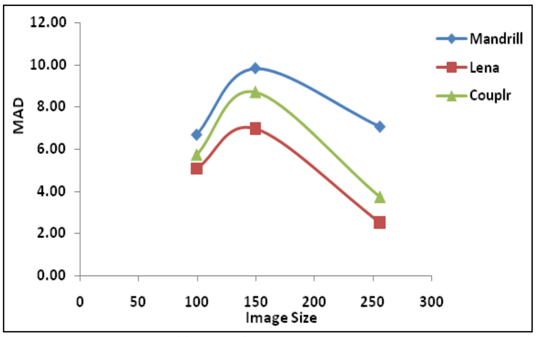


Fig (9) Effect of image size on MAD

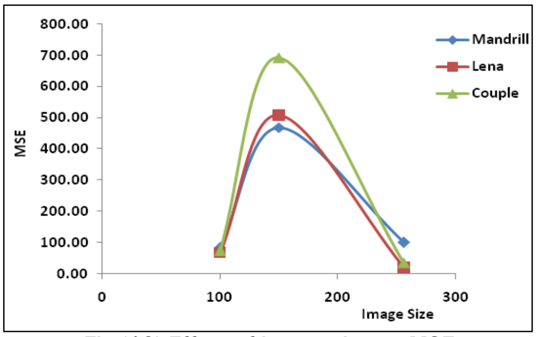


Fig (10) Effect of image size on MSE

6. Conclusions

The above listed results are conclude the following:

- **Encoding Time**

The ET decreases rapidly with the increase of image size for all three images .

- **PSNR**

The highest value of PSNR occurs when the image size equal to (256x256), but the lowest value of PSNR occurs when the image size equal to(150x150) for all three images, except in Mandrill image where we found that the highest value of PSNR occurs when the image size equal to (100x100).

- **CR**

CR is inversely proportional to image size, in another word CR decreases rapidly with the increase of image size for all three images.

- **Bit Rate, MAD, MSE**

From the table and diagrams we found the values of those measures are increase with increase of image size for all three images.

- **CF**

The values of CF are the same when the image size(100x100, 150 x 150) but it is increase when the image size is (256x256).

7.References

[LICH12]LIN ,Y., CHEN,W., " ***Fast Search Strategies for Fractal Image Compression***", Journal of Information Science and Engineering ,vol.28 ,pp.17-30, Taiwan ,2012.

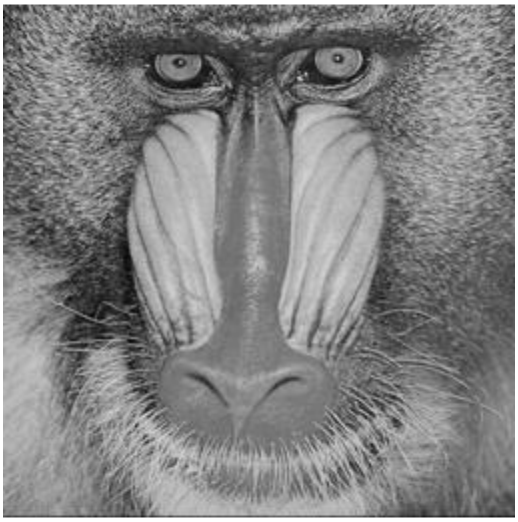
[Saa10]Saad,E., ***"Speeding Up Fractal Image Compression by Reducing Image Size.",*** Diyala Journal for pure Science, Vol: 6 No: 4, ISSN: 1992-0784, Baghdad, Iraq, October 2010.

[Hilo07] Al-Hilo, E.A., ***"Speeding-up Fractal Colored Image Compression using Moments Features*** ", PhD Thesis, College of Science, AL-Mustansiriyah University, Baghdad, 2007.

[Fish95] Fisher, Y., ***Fractal Image Compression Theory and Application***, Book, University of California, Institute for Nonlinear Science, Springer-Verlay, New York, Inc, 1995.

[GeHi11] George, L.E., Al-Hilo, E. A., ***"Color FIC by Adaptive Zero-Mean Method"*** International Symposium on Computing, Communication, and Control (ISCCC 2009) Proc .of CSIT vol.1 ,pp190-196, IACSIT Press, Singapore, 2011.

[ChRa09] Chakrapani, Y., Rajan, K. S." ***Genetic Algorithm Applied To Fractal Image Compression"***, VOL. 4, NO. 1,PP.53-58, February 2009.



(256x256)

(150x150) (100x100)

Mandrill

(150x150) (100x100)

Lena

(256x256)

(256x256)

(150x150) (100x100)

Couple