The Use of Quadtree Range Domain Partitioning with Fast Double Moment Descriptors to Enhance FIC of Colored Image

Bushra A. Sultan[1], Loay E. George[2], Nidaa F. Hassan[3]

[1]Department of Computer Science, College of Science, University of Baghdad, Baghdad - Iraq

[2]Department of Remote Sensing, College of Science, University of Baghdad, Baghdad - Iraq

[3]Department of Computer Science, University of Technology, Baghdad - Iraq

Abstract–**In this paper, an enhanced fractal image compression system (FIC) is proposed; it is based on using both symmetry prediction and blocks indexing to speed up the blocks matching process. The proposed FIC uses quad tree as variable range block partitioning mechanism. two criteria's for guiding the partitioning decision are used: The first one uses sobel-based edge magnitude, whereas the second uses the contrast of block. A new set of moment descriptors are introduced, they differ from the previously used descriptors by their ability to emphasize the weights of different parts of each block. The effectiveness of all possible combinations of double moments descriptors has been investigated. Furthermore, a fast computation mechanism is introduced to compute the moments attended to improve the overall computation cost. the results of applied tests on the system for the cases "variable and fixed range" block partitioning mechanism indicated that the variable partitioning scheme can produce better results than fixed partitioning one (that is, 4 × 4 block) in term of compression ratio, faster than and PSNR does not significantly decreased.**

Index Terms—**Fractal image compression, Iterated function system, Moments features, Quadtree.**

I. Introduction

Recently, fractal compression of digital images has attracted much attention. Fractal image compression (FIC) is based on the theory of iterated function system (IFS), and its performance relies on the presence of self-similarity degree (Mahadevaswamy, 2000). FIC process implies finding a set of transformations that produce fractal image which approximates the original image (Xi and Zhang, 2007). One of the most important characteristics of fractal image coding is its unsymmetrical property of encoding and decoding processing. Coding time is rather long for whole domain-range matching operation, whereas the decoding algorithm is relatively simple and fast. This weak aspect makes the fractal compression method not widely used as standard compression. FIC has the advantages of fast decompression as well as very high compression ratios (CR) (Al-Hilo and George, 2008).

Many attempts have been done for speeding FIC using different speeding-up methods. An adaptive zero-mean method was proposed by (George, 2006), according to this method the average of the range block is used instead of traditional offset parameter. George method's used in combination with moment-based features (George and Al-Hilo, 2008; Al-Hilo and George, 2008) and DCT-based methods (George and Minas, 2011) as IFS transform invariants to be used as block descriptors; which in turn is utilized to classify the domain and range blocks. Furthermore, by adding the symmetry predictor that introduced in the method given in (George and Al-Hilo, 2011) that based on using first-order centralized moments; this predictor is useful to reduce the number of isometric trails from (8, that is, Rotation, reflection…etc.,) trials to (1) trail. Mahmoud, 2012 proposed the use of double moment descriptors to speed up FIC.

In the proposed method, introduced in this paper, the loaded RGB color image was transformed to YUV color space, where Y is the luminance component, and U, V are the chromatic components. To get an effective compression, the U, V component are downsampled (Ning, 2007).Then, each component of YUV is coded individually using FIC method. An improved algorithm of FIC based on partition IFS method is applied; the improvement was in: (i) The scheme of range pool partitioning, (ii) IFS matching with low computation redundancy, and (iii) using a new set of centralized moments which are complementary and more informative.

The rest of the paper is structured as follows: Section II is dedicated to give an overview for the concepts and methods used to explain the enhance FIC system. The results of the tests applied the enhanced are discussed in Section III, and finally, the derived conclusions are listed in Section IV.

ARO-The Scientific Journal of Koya University
Volume VI, No.1 (2018), Article ID: ARO.10207, 10 pages
DOI: 10.14500/aro.10207
Received 24 February 2017; Accepted: 24 February 2018
Regular research paper: Published 24 March 2018
Corresponding author's, e-mail: bushraasultan@scbaghdad.edu.iq

 http://dx.doi.org/10.14500/aro.10207

II. Materials and methods

A. IFS Coding for Zero-mean Blocks

IFS coding based on zero-mean blocks matching implies that the offset values of the block replacement with average brightness values. Hence, the IFS mapping equation was performed according to this change. For a range block with pixel values (r_0,…..,r_{n-1}) and a domain block with pixels (d_0,…..,d_{n-1}) the contractive affine approximation is (George, 2006):

$$r_i' = s\left(d_i - \bar{d}\right) + \bar{r} \tag{1}$$

$$\bar{r} = \frac{1}{n}\sum_{i=0}^{n-1} r_i \tag{2}$$

$$\bar{d} = \frac{1}{n}\sum_{i=0}^{n-1} d_i \tag{3}$$

Where r_i' is the optimal approximated value of the ith byte value of the range block; d_i is the corresponding byte value in the best-matched domain block; s is the scaling coefficient; $\bar{d}, \bar{r}$ are the average of domain and range block, respectively.

To determine the scale (s) value, the method of least mean square errors (depicted in equation 1) is applied to get:

$$s = \begin{cases} \dfrac{\frac{1}{n}\sum_{i=0}^{n-1} d_i r_i - \bar{r}\bar{d}}{\sigma_d^2} & \text{if } \sigma_d^2 > 0 \\ 0 & \text{if } \sigma_d^2 = 0 \end{cases} \tag{4}$$

$$X^2 = \sigma_r^2 + s\left[s\sigma_d^2 + 2\bar{d}\bar{r} - \frac{2}{n}\sum_{i=0}^{n-1} d_i r_i\right] \tag{5}$$

Where,

$$\sigma_d^2 = \frac{1}{n}\sum_{i=0}^{n-1} d_i^2 - \bar{d}^2 \tag{6}$$

$$\sigma_r^2 = \frac{1}{n}\sum_{i=0}^{n-1} r_i^2 - \bar{r}^2 \tag{7}$$

At each range-domain matching instance, and before determination of χ^2 (equation 5), the scale coefficient (s) must be bounded to be in the range [$-s_{max}$, s_{max}]. Then, the scale coefficient (s) and $\bar{r}$ should be quantized using the following equations (Mahmoud, 2012):

$$\tilde{s} = Q_s I_s \tag{8}$$

$$I_s = \text{round}\left(\frac{s}{Q_s}\right) \tag{9}$$

$$\tilde{r} = Q_{\bar{r}} I_r \tag{10}$$

$$I_r = \text{round}\left(\frac{\bar{r}}{Q_{\bar{r}}}\right) \tag{11}$$

Where

$$Q_s = \frac{s_{max}}{2^{bs-1} - 1} \tag{12}$$

$$Q_{\bar{r}} = \frac{255}{2^{br} - 1} \tag{13}$$

Where s_{max} is the highest permissible value of the scale coefficient (s); Q_s and $Q_{\bar{r}}$ are the quantization steps of the scale and $\bar{r}$ coefficients, respectively; bs is the number of scale bits; br is the number of range mean bits.

B. Isometric Process Predictor

The eight isometric mappings are shown in Table I (George and Al-Hilo, 2009). A full search through the set of 8 isometric states of each block is prohibitive due to the large number of calculation involved. The goal is to exclude isometric states of blocks that have no chance of being selected as the best choice (George and Mahmoud, 2011).

The involved calculation for block indexing and transform prediction should be simpler than the full calculation. This would ease the burden of searching by reducing the set of possible candidates to minimal error. Hence, in this process, the speeding up of FIC is accomplished using the first-order moments descriptor (George and Al-Hilo, 2009). The theoretical basis of this predictor of the isometric processes is described in the following sections.

Centralized moments

For an image block *I(x,y){x,y| 0,1,….,L-1}*, its first-order centralized moments are defined as (George and Mahmoud, 2011):

$$M_x = \sum_{x=0}^{L-1}\sum_{y=0}^{L-1} I(x,y)(x-c) \tag{14}$$

$$M_y = \sum_{y=0}^{L-1}\sum_{x=0}^{L-1} I(x,y)(y-c) \tag{15}$$

Where c = (L−1)/2.

Combining equations (14) and (15) with the equations listed in Table I, the relationship between the new moments values (M'_x, M'_y) of a transformed block with its old moments' values (M_x, M_y), before transformation, could be determined; Table II shows these relationships (George and Al-Hilo, 2009).

Blocks classification

A method for blocks classification based on moment criteria is suggested. The classification is based on the status of its first-order moments values (that is, M_x and M_y). The following three status criteria have been used where:

- Condition-1: Is $|M_x| \geq |M_y|$ or not?
- Condition-2: Is $M_x \geq 0$ or not?
- Condition-3: Is $M_y \geq 0$ or not?

The use of these three Boolean criteria leads to eight block classes as illustrated in Table III (George and Al-Hilo, 2001).

At each range-domain matching instance, the indices of both domain and range blocks are passed through the predictor, as shown in Table IV. Then, the predictor outputs the index of the required isometric transform to get the best possible match between the domain and range blocks (George and Mahmoud, 2011).

C. Moment's Ratio and Index

Moment's ratio can be calculated using the following equation by (Mahmoud, 2012):

$$\mathrm{Ratio}_M = \begin{cases} \left|\dfrac{M_y}{M_x}\right| \times Nm & \text{if } M_x \geq M_y \\ \\ \left|\dfrac{M_x}{M_y}\right| \times Nm & \text{if } M_y \geq M_x \end{cases} \tag{16}$$

Where M_x and M_y are the moments around x and y coordinates, respectively. *Nm* is the maximum moment ratio value. Al-Hilo and George, 2008 concluded that "If the two blocks (range and domain) nearly satisfy the conductive affine transform, then their moment ratio factor ($Ratio_{Mr}$ and $Ratio_{Md}$) should have similar values, this does not mean that any two blocks have similar ($Ratio_M$) factor should necessarily satisfy the affine transform."

The value of combined moment ratio index is computed as the linear combination of two descriptors ($Ratio_{M1}$) and ($Ratio_{M2}$) using the following equation (Mahmoud, 2012).

$$I_M = [Ratio_{M1} \times (Nm+1) + Ratio_{M2}] \tag{17}$$

The index (I_M) is used to classify the domain and range blocks (that is, each class includes blocks having the same index). This factor is used to improve (that is, speeding up) the range-domain search task by only the domain blocks whose I_M values are similar (or near) to that of tested range block are IFS-matched.

D. The Proposed FIC

The aims of the enhanced FIC are:

- A new set of moment descriptors are introduced, they differ from the previously used ones by their excellent emphasis to reflect the moments' weight of certain part of the block. In this article, the effectiveness of all possible combinations of double moment descriptors had been investigated.
- Quadtree (QT) is used to enhance IFS performance. It is used as variable range blocks partitioning scheme instead of fixed block partitioning scheme. The criteria guiding the decomposition

TABLE I
ISOMETRIC TRANSFORMATION (GEORGE AND AL-HILO, 2009).

ID	Operation	Equation	Results
0	Identity	x' = x cos (0) - y sin (0) y' = – x sin (0) + y cos (0)	x' = x y' = y
1	Rotation (+90)	x' = x cos (90) - y sin (90) y' = –x sin (90) + y cos (90)	x' = y y' = –x
2	Rotation (+180)	x' = x cos (180) - y sin (180) y' = –x sin (180) + y cos (180)	x' = –x y' = –y
3	Rotation (+270)	x' = x cos (270) - y sin (270) y' = –x sin (270) + y cos (270)	x' = –y y' =x
4	Reflection around X-axis	x' = –x cos (0) - y sin (0) y' = –x sin (0) + y cos (0)	x' = –x y' =y
5	Reflection around X-axis+Rotation(+90)	x' = –x cos (90) - y sin (90) y' = –x sin (90) + y cos (90)	x' = –y y' = –x
6	Reflection around X-axis+Rotation(+180)	x' = –x cos (180) - y sin (180) y' = –x sin (180) + y cos (180)	x' =x y' = –y
7	Reflection around X-axis+Rotation(+270)	x' = –x cos (270) - y sin (270) y' = –x sin (270) + y cos (270)	x' =y y' =x

TABLE II
THE RELATIONSHIP BETWEEN MOMENTS BEFORE AND AFTER APPLYING THE ISOMETRIC TRANSFORMATION (GEORGE AND AL-HILO, 2009)

ID	Operation	Relationship
0	Identity	$M'_x=M_x$, $M'_y=M_y$
1	Rotation (+90)	$M'_x=M_y$, $M'_y=-M_x$
2	Rotation (+180)	$M'_x=-M_x$, $M'_y=-M_y$
3	Rotation (+270)	$M'_x=-M_y$, $M'_y=M_x$
4	Reflection at X-axis	$M'_x=-M_x$, $M'_y=M_y$
5	Reflection around X-axis+rotation (+90°)	$M'_x=-M_y$, $M'_y=-M_x$
6	Reflection around X-axis+rotation (+180°)	$M'_x=M_x$, $M'_y=-M_y$
7	Reflection around X-axis+rotation (+270°)	$M'_x=M_y$, $M'_y=M_x$

TABLE III
THE TRUTH TABLE FOR EIGHT BLOCK CLASSES (MAHMOUD, 2012)

Block class ID	Boolean criteria		
	$\lvert M_x\rvert \geq \lvert M_y\rvert$	$M_x \geq 0$	$M_y \geq 0$
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 - Identity, 1 - Rotation (+90), 2 - Rotation (+180), 3 – Rotation (+270), 4 - Reflection, 5 - Reflection+rotation (+90), 6 - Reflection+Rotation (+180), 7 - Reflection+Rotation (+270).

TABLE IV
THE REQUIRED ISOMETRIC OPERATION TO CONVERT THE BLOCK STATE (GEORGE AND MAHMOUD, 2011)

Range blocks ID	Domain blocks ID							
	0	1	2	3	4	5	6	7
0	0	6	4	2	7	3	1	5
1	6	0	2	4	1	5	7	3
2	4	2	0	6	3	7	5	1
3	2	4	6	0	5	1	3	7
4	7	3	1	5	0	6	4	2
5	1	5	7	3	6	0	2	4
6	3	7	5	1	4	2	0	6
7	5	1	3	7	2	4	6	0

0 - Identity, 1 - Rotation (+90), 2 - Rotation (+180), 3 – Rotation (+270), 4 - Reflection, 5 - Reflection+Rotation (+90), - Reflection+Rotation (+180), 7 - Reflection+Rotation (+270).

http://dx.doi.org/10.14500/aro.10207

process is the information richness of the region; it was used to decide the initial partitioning of the range blocks.

- FIC algorithm is reconfigured including the moment equations to remove any redundancy in the computation.

In the following subsection, the proposed enhanced FIC is explained in more details.

The proposed moments and the speeding up mechanism

A new set of weights is introduced and adopted to produce the new sets of moments, they are as follows:

$$W_1(i) = \begin{cases} \frac{2}{L}\left(i - \frac{L}{2}\right) & \text{for } i = \left[0, \frac{L}{2}\right) \\ -W_1(L-1-i) & \text{for } i = \left[\frac{L}{2}-1, L\right) \end{cases} \quad (18\text{-a})$$

$$W_1^{int}(i) = W_1(i) \times 100 \quad \text{for } i = [0, L) \quad (18\text{-b})$$

$$W_2(i) = \begin{cases} \frac{2}{L}\left(i - \frac{1}{2}\right) & \text{for } i = \left[0, \frac{L}{2}\right) \\ -W_2(L-1-i) & \text{for } i = \left[\frac{L}{2}-1, L\right) \end{cases} \quad (19\text{-a})$$

$$W_2^{int}(i) = W_2(i) \times 100 \quad \text{for } i = [0, L) \quad (19\text{-b})$$

$$W_3(i) = \begin{cases} \sin\left(\frac{\Pi i}{L-1}\right) & \text{for } i = \left[0, \frac{L}{2}\right) \\ -W_3(L-1-i) & \text{for } i = \left[\frac{L}{2}-1, L\right) \end{cases} \quad (20\text{-a})$$

$$W_3^{int}(i) = W3(i) \times 100 \quad \text{for } i = [0, L) \quad (20\text{-b})$$

Fig. 1 presents the proposed three weight functions that assigned to each row or column within the block, when its length is equal to 8.

The new sets of moments around x-axis and around y-axis that using the weights functions given in equations (18-20) are defined as:

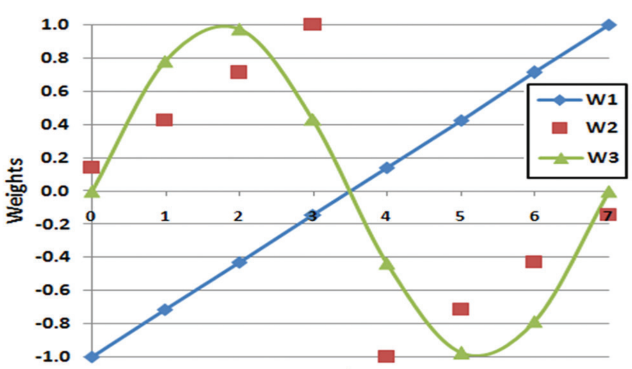

Fig. 1. The proposed weights.

$$Mx_1(x,y) = \sum_{r=x}^{r+L-1}\sum_{c=y}^{c+L-1} f(r,c)\,W_1^{int}(r-x) \quad (21\text{-a})$$

$$My_1(x,y) = \sum_{c=y}^{c+L-1}\sum_{r=x}^{r+L-1} f(r,c)\,W_1^{int}(c-y) \quad (21\text{-b})$$

$$Mx_2(x,y) = \sum_{r=x}^{r+L-1}\sum_{c=y}^{c+L-1} f(r,c)\,W_2^{int}(r-x) \quad (22\text{-a})$$

$$My_2(x,y) = \sum_{c=y}^{c+L-1}\sum_{r=x}^{r+L-1} f(r,c)\,W_2^{int}(c-y) \quad (22\text{-b})$$

$$Mx_3(x,y) = \sum_{r=x}^{r+L-1}\sum_{c=y}^{c+L-1} f(r,c)\,W_3^{int}(r-x) \quad (23\text{-a})$$

$$My_3(x,y) = \sum_{c=y}^{c+L-1}\sum_{r=x}^{r+L-1} f(r,c)\,W_3^{int}(c-y) \quad (23\text{-b})$$

Where L represents the length of the block; (x,y) are coordinates of the block relative to left-top corner; $F()$ is the 2D image array; W^{int} represents the integer index of the weights; Mx represents the low-order moments relative to x-axis; My represents the low-order moments relative to y-axis.

So, for each block, of the overlapped blocks listed in the domain pool, the moments (given by equations 21-23) have been computed. For fast computations of the moment descriptor equations the following scenario adopted to avoid the redundant summation occurs within each one by creating two 2D arrays, named "SumX" and "SumY," such that:

$$SumX(x,0) = \sum_{y=0}^{L-1} f(x,y) \quad (24)$$

$$SumX(x,y) = SumX(x,y-1) - f(x,y-1) + f(x,y+L-1) \quad (25)$$

$$SumY(0,y) = \sum_{x=0}^{L-1} f(x,y) \quad (26)$$

$$SumY(x,y) = Sumy(x-1,y) - YB(x-1,y) + YB(x+L-1,y) \quad (27)$$

Fig. 2 shows simple example of how the value of *SumX* (0,0) and *SumX* (0,1) had been computed for row sample when block length (*L*) equal to 8.

Now, the created arrays *SumX* and *SumY* can be used in (21-23) to become as:

$$Mx_1(x,y) = \sum_{r=x}^{r+L-1} SumX(r,y) \times W_1^{int}(r-x) \quad (28a)$$

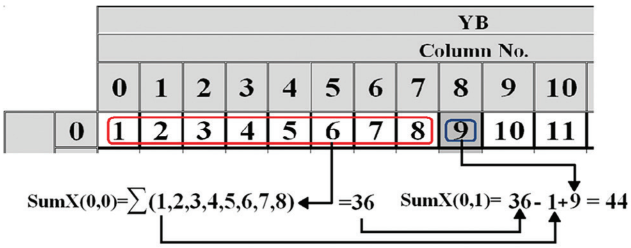

Fig. 2. Shows simple example of how the value of *SumX* (0,0) and *SumX* (0,1) had been computed for row sample when block length (L) equal to 8.

$$My_1(x,y)=\sum_{c=y}^{y+L-1} SumY(x,c)\times W_1^{int}(c-y) \tag{28b}$$

$$Mx_2(x,y)=\sum_{r=x}^{r+L-1} SumX(r,y)\times W_2^{int}(r-x) \tag{29a}$$

$$My_2(x,y)=\sum_{c=y}^{y+L-1} SumY(x,c)\times W_2^{int}(c-y) \tag{29b}$$

$$Mx_3(x,y)=\sum_{r=x}^{r+L-1} SumX(r,y)\times W_3^{int}(r-x) \tag{30a}$$

$$My_3(x,y)=\sum_{c=y}^{y+L-1} SumY(x,c)\times W_3^{int}(c-y) \tag{30b}$$

As mentioned above, the effectiveness of using the three possible pairs of moments combinations, to compute the blocks descriptors that used for indexing to speeding up the range-domain search task, {that is, (M_1,M_2), (M_1,M_3), and (M_2,M_3)} have been investigated.

The proposed range pool partitioning scheme

The proposed partitioning scheme that applied to generate the range pool blocks is the quadtree; it partitions the range array into nonoverlapped variable length blocks. Two criteria have been used to guide the decomposition process; the first is based on the edge detection using Sobel filter (see equations 31a and b) and the second based on the contrast (see equation 32). The permissible block length (PBL) is 8 and 4:

$$G_x=\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix} \tag{31a}$$

$$G_y=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \tag{31b}$$

For the block addressed as $\begin{bmatrix}(x,y) & (x,y+1)\\ (x+1,y) & (x+1,y+1)\end{bmatrix}$ the maximum variance around the pixel (x,y) is computed as

$$Max_{var}=Max\begin{pmatrix}\left|pixel(x,y)-pixel(x,y+1)\right|,\\ \left|pixel(x,y)-pixel(x+1,y)\right|,\\ \left|pixel(x,y)-pixel(x+1,y+1)\right|\end{pmatrix} \tag{32}$$

The regions threshold determined according to the following relation:

$$Thr_Q=\begin{cases} val_{var} & \text{if use } Max_{var}\\ val_{Sobel} & \text{if use } Max_{Sobel}\end{cases} \tag{33}$$

Where $val_{var}\in\{10,12,14.,30\}$, while $val_{sobel}\in\{30,35,40,\ldots,90\}$. These threshold values were selected after comprehensive tests.

The partitioning process started by partitioning the range array into nonoverlapped block of size equal to (8). For each block, one of the following steps is done:

- If Sobel filter used as a decision to partition the current block, then the highest value between G_x or G_y over the tested block determined,
- If the maximum variance is used as a decision to partition this block, then the maximum variance over all pixels belong to the block are determined.

If the determined value exceeds the predetermined Thr_Q, then split the block into four quadrants.

Fig. 3a shows the well-known Baboon image after applying the proposed partitioning scheme that used Sobel as partitioning decision (where the threshold value val_{sobol} is set 40). While Fig. 3b shows the well known Lenna image after applying the proposed partitioning scheme that based on maximum variance as partitioning decision (where the threshold value $val_{variance}$ is set to 22).

Enhanced FIC encoding process

The introduced enhanced encoding algorithm of the range blocks are summarized by the following steps:

a) Load BMP image and put it in (R,G,B) array (three 2D arrays)
b) Convert (R,G,B) array to (Y,U,V) array
c) Downsample the components U and V
d) Determine the moment combinations used
e) For each color component (that is, the original Y, and the downsampled [U,V]) do the followings:
   1. Construct the domain and range pools. This process is done by partitioning the range array into nonoverlapping variable blocks (using the above-mentioned quadtree method with one of the partitioning decision stated in section 2.4.2) to generate the range $(r_0,..r_{n-1})$ blocks.
   2. Set the PBL to 8.
   3. The downsampled range array (that is, the domain array) is partitioned into overlapped fixed blocks of size equal to PBL to generate the domain $(d_0,..d_{m-1})$ blocks.
   4. For each domain block listed in domain pool do the following:
   i. Determine the average $(\bar{d})$ using equation 3.
   ii. Determine the $(n\sigma_d^2)$ as:

$$n\sigma_d^2 = \sum_{i=0}^{n-1} d_i^2 - n\bar{d}^2 \tag{34}$$

iii. Determine the moments using equations (28-30) that corresponding to the selected moments.
iv. Determine the moment ratio for each moment using equation 16.
v. Determine the moment index value using equation 17.
vi. Determine the block isometric index (*Sym_indxd*) that based on moment order one (that is, equations (14,15) and Table III).

5. Store the position coordinates (x_d, y_d) of the domain blocks and the calculated moment index in temporary array (L) of record.
6. Sort the array of record (L) in ascending order according to their moment index.
7. Establish a set (P) of pointers referring to the start and end of records holding the same index value.
8. For each range, block of size equal to PBL do the following:

i. Calculate the average $(\bar{r})$ using equation 2.
ii. Determine the $(n\sigma_d^2)$ as:

$$n\sigma_r^2 = \sum_{i=0}^{n-1} r_i^2 - n\bar{r}^2 \tag{35}$$

iii. Determine the moments using equations (28-30) that corresponding to the selected moments.
iv. Determine the moment ratio for each moment using equation 16.
v. Determine the moment index value using equation 17.
vi. Determine the block isometric index (*Sym_indxr*) that based on moment order one (that is, equations (14, 15) and Table III).

9. With help of pointers set (p) and the temporary list of records (L); match only the domain blocks whose moment index values equal to the range index value.
10. The parameters *Sym_indxr* and *sym_indxd* are passed through the isometric predictor; the predictor will output the index of the required isometric transform (using Table IV). Then, apply the assigned transform on the tested range block.
11. The proposed enhanced steps for fast computation of s and χ^2 is applied using the following equations:

$$\phi = \sum_{i=0}^{n-1} r_i d_i - n\bar{r}\bar{d} \tag{36}$$

$$s = \frac{\phi}{n\sigma_d^2} \tag{37}$$

$$\chi^2 = n\sigma_r^2 + s^2 n\sigma_d^2 - 2s\phi \tag{38}$$

Then, calculate the scale coefficient (s) and (χ^2) of first class of the domain blocks with range block and apply the following steps:

i. Compare the result (χ^2) of each matching instance with the (χ^2_{min}) that registered during the previous matching instances. If the compared (is smaller than (χ^2_{min}), then put its value in (χ^2_{min}) and register (s) beside to the average mean value of range block and the position and symmetry state (sym) of the domain block.
ii. If the (χ^2_{min}) < *minimum blockerror* , then the search across the domain blocks is stopped and the registered domain blocks are considered as the best match, output the set values (position, symmetry state, s, $\bar{r}$) and go to 4.
iii. Otherwise, start test the domain blocks that closest higher class.

12. Divide PBL by 2, if the result equal to 2 then, stop the search, else go to 3.

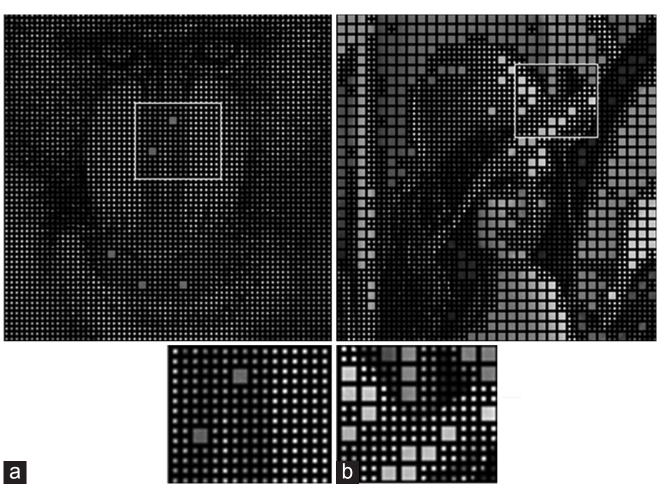

Fig. 3. (a and b) The proposed variable partitioning scheme applied on some test images.

III. Test results

The proposed system was implemented using Embarcadero RAD Studio 2010 Visual Pascal Programming Language programming language. The tests were conducted under the environment: Windows-8 operating system, laptop computer - Lenovo (processor: Intel (R) Core(TM) i5-3337U, CPU 1.8 GHz, and 4GB RAM). The tests were applied on the well-known Lena and Baboon image samples (whose specifications are: Size = 256 × 256 pixel, color depth = 24 bit). To assess the difference between the reconstructed image and the original images, the error measures (that is, mean square error MSE and peak signal

TABLE V
The Values of the Control Parameters

Parameter	Range or value
S_{max}	3
bs	6
br	8
Minimum Block Error	1
$Val_{variance}$	{10,12,14,.,30}
Val_{Sobel}	{30,35,40,.,90}
Nm	{25,30,35,.,50}

to noise ratio PSNR measured in dB) were used. Beside these fidelity measures, some complementary measures were used to describe the performance of the system; both the CR and bit rate parameters (BR) were used to describe the compression gain.

Table V lists the considered control parameters (including their names and default values); these values were selected after making comprehensive tests.

Table VI lists the notations with their descriptions that are used in the figures and tables presented in this section.

A. Moment Combination Test

In this set of tests, the effects of using the different combinations of proposed moments are illustrated, when the blocks have been partitioned fixedly to (4×4). (that is, PFIC4**)** as listed in Tables VII and VIII.

The parameters values listed the shaded rows in Tables VII and VIII were adopted to study the effect of the proposed moment combinations' on (CR, ET, and PSNR) with the test samples explained in Fig. 4.

From the listed results, it is obvious that T1 is the best-balanced combination. Since if the combinations sorted according to the maximum CR, the result is (T2, T0, and then T1), and the lowest attained ET result is (T1, T0, and then T2). It is difficult to arrange (T0, T1, and T2) according to the PSNR because there is no significant difference with test images.

B. PFIC4 and PFIC8 Versus PFICS Test

In this section, the proposed FIC was applied with quadtree partitioning scheme that using Sobel filter as partitioning decision (PFICS). The tests were conducted to investigate the effect of the proposed quadtree partitioning with moment combination versus fixed block partitioning (PFICL); the value of L is set equal to (4 and 8) with same moment combinations listed in Table IX. Fig. 5 illustrates

TABLE VI
The Notations Used in the Test Results

Notation	Description
T0	The moment index ratio computed from the combination of M1 and M2
T1	The moment index ratio computed from the combination of M1 and M3
T2	The moment index ratio computed from the combination of M2 and M3
PFIC4	The proposed FIC applied with fixed block partitioning of size equal to 4
PFIC8	The proposed FIC applied with fixed block partitioning of size equal to 4
PFICS	The proposed FIC applied with quadtree partitioning scheme that used Sobel as partitioning decision
PFICV	The proposed FIC applied with quadtree partitioning scheme that used variance as partitioning decision

TABLE VII
Test Results of Pfic4 When Applied on Lena Image

Moments combination	Nm	CR	MSE	PSNR	BR	ET (s)
T0	25	8.021	94.062	28.397	2.992	0.313
	30	8.027	8.021	28.262	2.990	0.235
	35	7.963	94.106	28.395	3.014	0.252
	40	7.964	97.966	28.220	3.013	0.244
	45	7.967	100.526	28.108	3.012	0.231
	50	7.980	106.446	27.860	3.008	0.235
T1	25	8.027	95.281	28.341	2.990	0.250
	30	8.014	100.351	28.116	2.995	0.192
	35	7.964	95.647	28.324	3.014	0.212
	40	7.965	99.035	28.173	3.013	0.218
	45	7.974	102.785	28.012	3.010	0.208
	50	7.977	107.596	27.813	3.009	0.209
T2	25	8.026	90.288	28.575	2.990	0.324
	30	8.032	96.783	28.273	2.988	0.247
	35	7.964	92.078	28.489	3.013	0.268
	40	7.969	98.046	28.217	3.012	0.259
	45	7.981	100.263	28.119	3.007	0.256
	50	7.983	103.335	27.988	3.006	0.254

TABLE VIII
Test Results of Pfic4 When Applied on Baboon Image

Moments combination	Nm	CR	MSE	PSNR	BR	ET (s)
T0	25	8.017	489.506	21.233	2.994	0.221
	30	7.984	499.533	21.145	3.006	0.199
	35	7.948	516.252	21.002	3.020	0.183
	40	7.952	519.355	20.976	3.018	0.194
	45	7.952	529.911	20.889	3.018	0.183
	50	7.952	549.678	20.730	3.018	0.182
T1	25	8.016	493.163	21.201	2.994	0.204
	30	7.985	503.922	21.107	3.006	0.187
	35	7.947	518.631	20.982	3.020	0.176
	40	7.950	521.902	20.955	3.019	0.188
	45	7.952	535.173	20.846	3.018	0.176
	50	7.951	549.090	20.734	3.019	0.179
T2	25	8.025	484.939	21.274	2.991	0.259
	30	7.990	497.737	21.161	3.004	0.228
	35	7.952	506.847	21.082	3.018	0.215
	40	7.951	513.062	21.029	3.018	0.219
	45	7.951	523.084	20.945	3.019	0.211
	50	7.952	539.883	20.808	3.018	0.207

TABLE IX
Test Results for Pfics and Pfics When the Moment Combination is t1, nm=50, and the Tested Sample is Baboon Image

A. PFICS results

Val_{Sobel}	CR	MSE	PSNR	BR	ET (sec)
30	8.160	568.479	20.584	2.941	0.116
35	8.401	571.162	20.563	2.857	0.110
40	8.603	573.228	20.548	2.790	0.105
45	8.774	574.969	20.534	2.735	0.105
50	9.033	578.903	20.505	2.657	0.101
55	9.247	580.188	20.495	2.595	0.098
60	9.432	584.204	20.465	2.544	0.096
65	9.608	585.074	20.459	2.498	0.095
70	9.752	587.470	20.441	2.461	0.095
75	9.930	590.511	20.419	2.417	0.096
80	10.060	593.388	20.397	2.386	0.093
85	10.292	599.283	20.355	2.332	0.093
90	10.477	604.115	20.320	2.291	0.090

B. PFIC4 and 8 results

L	CR	MSE	PSNR	BR	ET (s)
4	7.951	549.090	20.734	3.019	0.179
8	31.829	844.843	18.863	0.754	0.053

http://dx.doi.org/10.14500/aro.10207

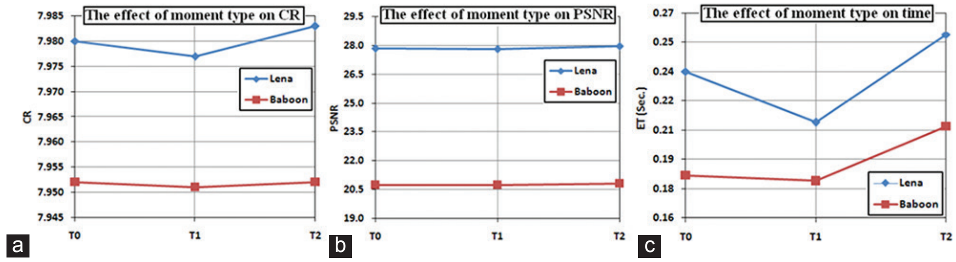

Fig. 4. (a-c) The effect of moment type with test images.

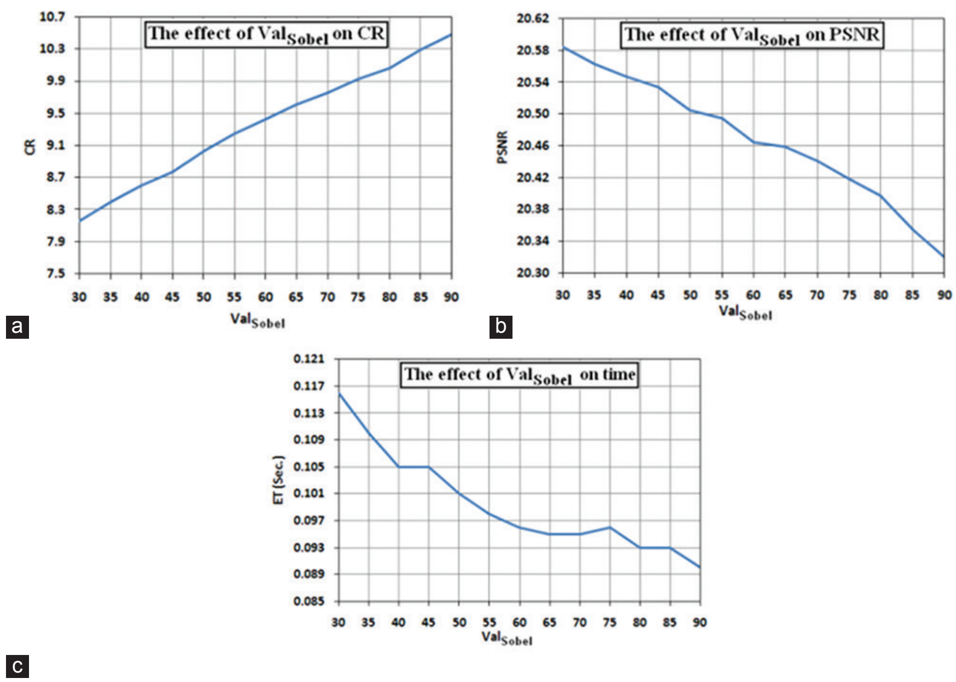

Fig. 5. (a-c) The effect Val_{Sobel}.

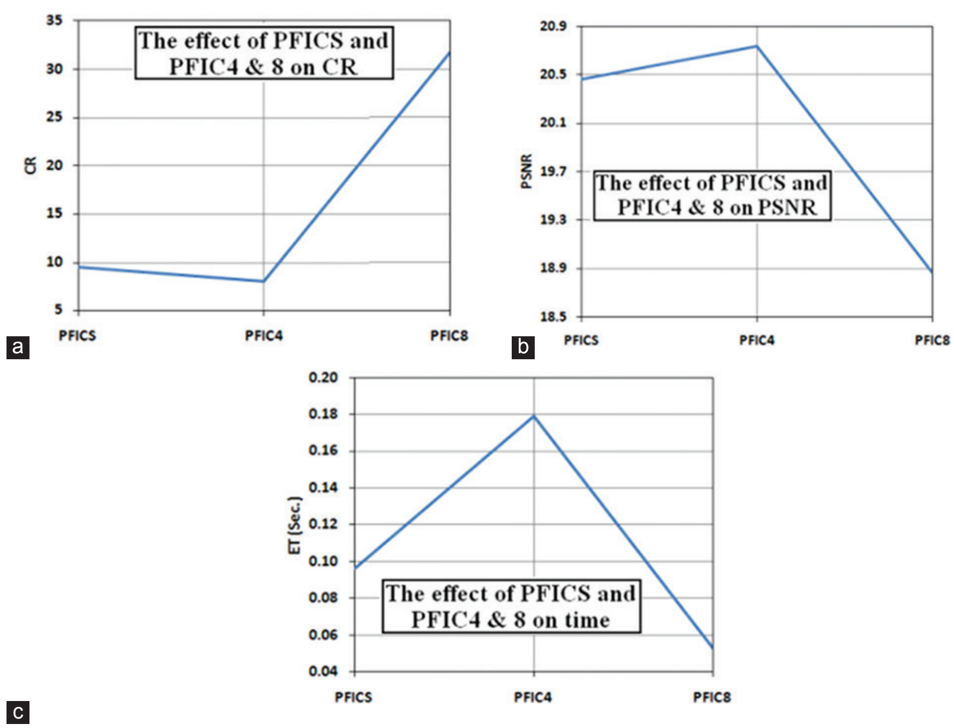

Fig. 6. (a-c) The effect of proposed variable partitioning scheme using Sobel with fixed one.

the effect of Val_{Sobel} on CR, ET, and PSNR when PFICS had been applied.

The parameters values in the listed shaded rows of Table IX have been used to compare the performance of variable and fixed partitioning scheme; the attained results are presented in Fig. 6.

C. PFIC4 and PFIC8 Versus PFICV Test

In this section, the proposed FIC was applied with quadtree partitioning scheme that using max variance as partitioning decision (PFICV). The tests were conducted to investigate the effect of the proposed quadtree partitioning with moment combination versus (PFICL) fixed block partitioning; the value of L is set to (4 and 8) with the same moment combinations listed in Table X. Fig. 7 illustrates the effect of $Val_{variance}$ on CR, ET and PSNR when PFICV was applied.

The parametric values of the shaded rows listed in Table X have been used to extract the results shown in Fig. 8 to compare the performance of variable and fixed partitioning scheme.

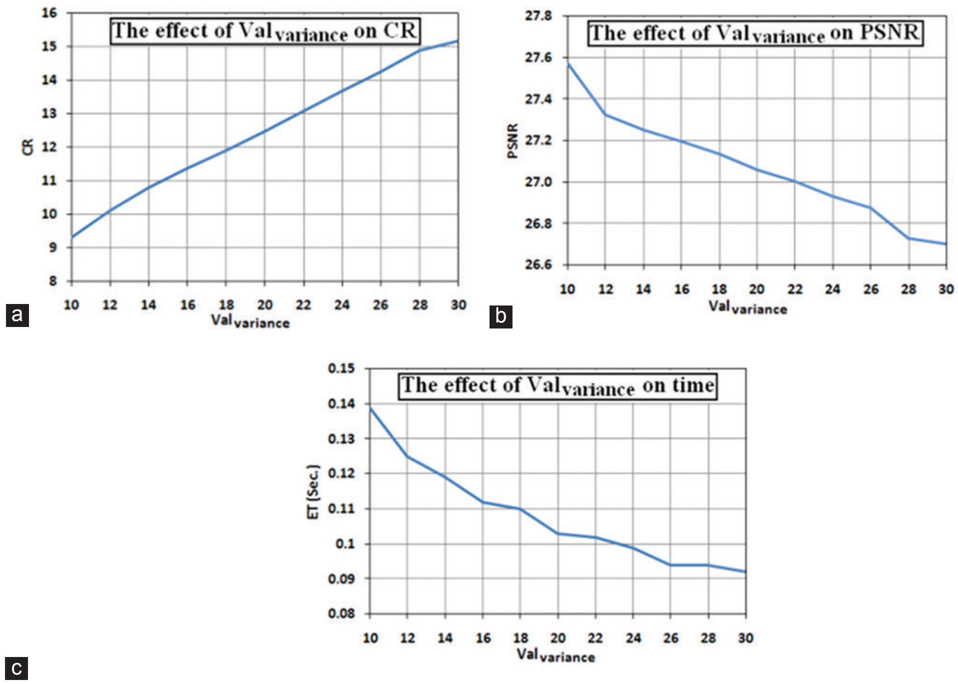

Fig. 7. (a-c) The effect $Val_{variance}$.

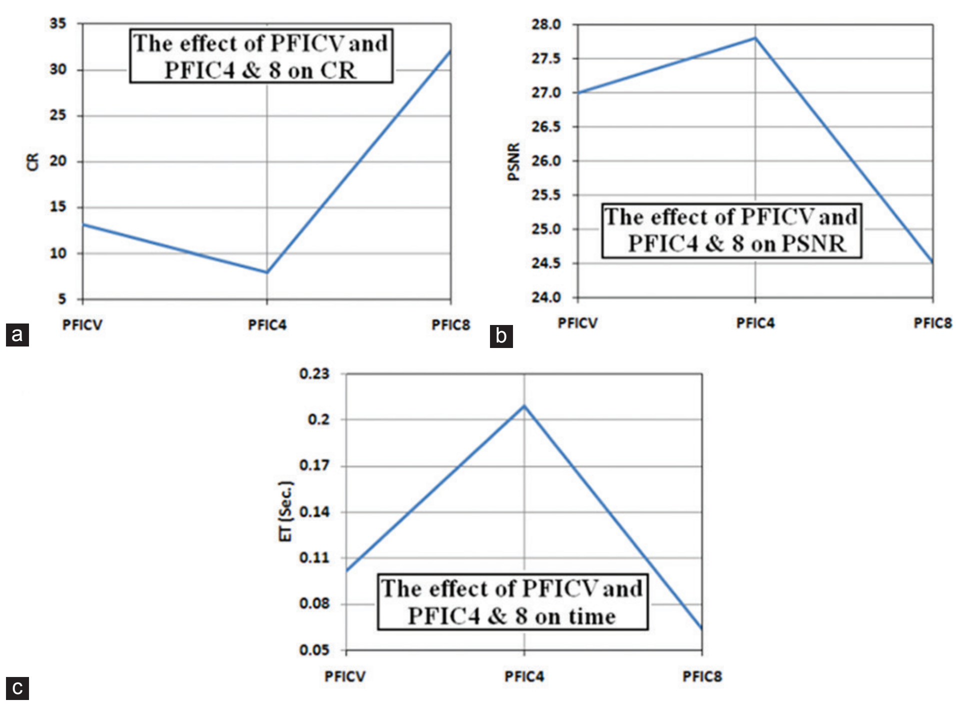

Fig. 8. (a-c) The effect of the proposed variable partitioning scheme using max variance with fixed one.

Table X.
Pfics test when moment combination is t1, nm=50 and the test image is lena

A. PFICV results					
$Val_{variance}$	CR	MSE	PSNR	BR	ET (sec)
10	9.321	113.734	27.572	2.575	0.139
12	10.108	120.344	27.327	2.374	0.125
14	10.803	122.395	27.253	2.222	0.119
16	11.377	123.949	27.198	2.110	0.112
18	11.919	125.725	27.137	2.014	0.110
20	12.497	128.012	27.058	1.921	0.103
22	13.095	129.634	27.004	1.833	0.102
24	13.705	131.865	26.930	1.751	0.099
26	14.280	133.470	26.877	1.681	0.094
28	14.886	138.039	26.731	1.612	0.094
30	15.184	138.930	26.703	1.581	0.092
B. PFIC4 and 8 results					
L	CR	MSE	PSNR	BR	ET (s)
4	7.977	107.596	27.813	3.009	0.209
8	32.115	229.912	24.515	0.747	0.064

IV. Conclusions

From the results of tests conducted on the proposed system, the following remarks were stimulated:

1. The quadtree partitioning that guided by Sobel or variance can be considered as useful partitioning mechanism.
2. The proposed FIC that uses double moment with variable partitioning schemes produces better results than that of fixed partitioning as shown in Figs. 6 and 8.
3. The proposed moments are suitable to perform double block description which in turn can significantly speed up the IFS encoding process.
4. For future work:
    a. A new criteria could be adopted for quadtree partitioning.
    b. Investigate the combination of triple moments.

References

Al-Hilo, E.A. and George, L.E., 2008. Speeding-up fractal colored image compression using moments features. IEEE, Digital Image Computing: Techniques and Applications, DICTA, 2008, pp.486-490.

George, L.E and Minas, N.A., 2011. Speeding up fractal image compression using DCT descriptors. Journal of Information and Computing Science, 6, pp.287-294.

George, L.E. and Al-Hilo, E.A, 2011. Speeding-up Fractal Color Image Compression Using Moments Feature Based on Symmetry Predictor, *IEEE, Computer Society, Eighth International Conference on Information Technology: New Generations*, Las Vegas, NV, pp. 508-513.

George, L.E. and Al-Hilo, E.A., 2008. Speeding-up Fractal Colored Image Compression Using Moments Features, *International Conference on Computer and Communication Engineering*, pp. 1303-1307.

George, L.E. and Al-Hilo, E.A., 2009. Fractal Color Image Compression by Adaptive Zero-Mean Method, IEEE Computer Society, *International Conference on Computer Technology and Development*, pp.525-529.

George, L.E. and Al-Hilo, E.A., 2009. Speeding-Up Color FIC Using Isometric Process Based on Moment Predictor, *International Conference on Future Computer and Communication, IEEE Computer Society,* pp.607-611.

George, L.E. and Mahmoud, S.L., 2011. Image steganography using an accelerated affine Block matching scheme. International Journal of Computer Information Systems, 2, pp.1-7.

George, L.E., 2006. IFS coding for zero-mean image blocks, university of Baghdad, college of science, Iraqi. Journal of Science, 47, pp. 190-194.

Mahadevaswamy, H.R., 2000. *New Approaches to Image Compression, Ph.D. Thesis, Regional Engineering College,* University of Calicut.

Mahmoud, S.L., 2012. The use of double moment based descriptors to speed up FIC. Journal of Al-Nahrain University, Science, 15, pp. 200-202.

Ning, L., 2007. *Fractal Imaging, a Book.* Academic Press, New York.

Xi, L. and Zhang, L., 2007. A study of fractal image compression based on an improved genetic algorithm. International Journal of Nonlinear Science, 3, pp.116-124.