Eng. & Tech. Journal, Vol.28 No.10, 2010

New Technique For Reducing Symmetry Mapping In Colored FIC Based on Moments Features

Dr. Abdul-Monem S.Rahma* & Dr. Raheem A. S. Ugla*

Received on:16/11/2009
Accepted on:16/2/2010

Abstract

This paper studded the effect of symmetry mapping process on the compression parameters of the fractal color image compression by moment features was studded. Feature of moment utilized to reduction the symmetry mapping from 8 to only one. The operation of reduction is achieved by using predictor to symmetry mappings; the predictor will predict specific symmetry mapping according a specific feature of moments to one of eight. Such that eight versions (blocks) are produced for each domain block, so this case needs 8 mappings and it requires more computational time. Our suggestion will directly reduce the encoding time 1:8 times.

Keywords: frame Compression, Fractal Image Compression, Isometric Process, Moments Features

تقنية جديدة لتقليل عماليات التناظر في ضغط الصور بأستخدام الكسوريات

الخلاصة

يدرس هذا البحث تأثيرعملية تحويل التنـــاظر(symmetry mapping process) علـــى متغيرات الضغط الخاصة بضغط الصور الملونة باستخدام الكسوريات(fractals) أعتمادا" على خصائص العزوم. حيث أستخدمت خصائص العزوم لتقليل عمليات التناظر من ثمان حالات الى حالة واحدة. عملية التقليل هذه أنجزت بتصميم متنبأ (predictor) لعمليات التناظر. وظيفة هذا المتنبأ يتنبأ بعملية تحويل محددة وفقا" لخاصية محددة من خصائص العزوم وهي واحدة من ثمان حالات. حيث هنالك ثمان حالات تناظر لكل كتلة (Block) تُعرف لكل كتلة فضاء (Domain Block) .وهذة الحالات تحتاج الى ثمان تحويلات مما يودي الى استهلاك وقت كبير لأجـــراء العمليات الحسابية المطلوبة لأجراء مطابقة بين كتلة المدى(Range Block) وكتلـــة الفضـــاء (Domain Block) . البحث المقترح الذي تم تنفيذه يقلل زمن الضغط بنسبة (1:8) تـــم التنفيـــذ علـــى حاســـبة شخصـــية(Pentium IV, 1.79 GHz, 224 MB or RAM) Windows XP, ، باستخدام (VB 6)

***Computer Science Department, University of Technology/ Baghdad**

https://doi.org/10.30684/etj.28.10.12
2412-0758/University of Technology-Iraq, Baghdad, Iraq

1. Introduction

The field of fractal coding of still image has become more and more popular among the image coding community. The initial works of Barnsly [1] and jacquin [2] were based on self-similarity within images. The image to be coded is partitioned into non overlapping blocks (called range blocks), each of them being then described as a contractive transformation is made of a set of isometrics and a contraction of the luminance profile to which a constant is added. Saup [3] represents selected studies about the influence of isometrics on the quality of fractal codes. Schebe [4] analysis's the position of isometrics in fractal transformation.

The aim of this paper is to studding the behavior of isometric process and reducing them from 8 to only one in the fractal color image compression using moment's features.

In this work the loaded RGB color image was transformed to YUV color space,. In order to get an effective compression, Where the (U, V) component are down sampled [5]. Each component of YUV was coded individually using FIC (Fractal Image Coding) method.

2. Iterated Function System(IFS) Coding for Zero-Mean Blocks

The basic idea of PIFS (Partition IFS) is partitioning the image into non overlapping range blocks. For every range block a similar but larger domain block is found. This research uses a fixed size partitioning scheme, since it requires less computational time than the other [6].

For a range block with pixel values (r_0, r_1,…,r_{m-1}), and domain block (d_0, d_1,…,d_{m-1}) the contractive affine approximation of zero-mean blocks is [7]:

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

Where, the fractal parameters become scale (s) and range average ($\bar{r}$) instead of the conventional scale (s) and offset (o) coefficients in traditional IFS mapping equation.

The scale parameter (s) could be determined by applying the least mean square difference (χ^2) between the approximated (r_i') and actual (r_i) value [7]:

$$\chi^2 = \frac{1}{m}\sum_{i=0}^{m-1}\left(r_i' - r_i\right)^2 \qquad \ldots.(2)$$

$$\frac{\partial \chi^2}{\partial s} = 0 \qquad \ldots. (3)$$

A straight forward manipulation for equations (2, 3) leads to:

$$s = \begin{cases} \dfrac{\frac{1}{m}\sum_{i=0}^{m-1} d_i r_i - \bar{r}\bar{d}}{\sigma_d^2} & if\ \sigma_d^2 > 0 \\ 0 & if\ \sigma_d^2 = 0 \end{cases} \qquad \ldots(4)$$

$$\chi^2 = \sigma_r^2 + s\left[s\sigma_d^2 + 2\bar{d}\bar{r} - \frac{2}{m}\sum_{i=0}^{m-1} d_i r_i \right] \qquad \ldots. (5)$$

Where,

$$\sigma_d{}^2 = \frac{1}{m}\sum_{i=0}^{m-1} d_i{}^2 - \bar{d}^2 \qquad \ldots(6)$$

$$\sigma_r{}^2 = \frac{1}{m}\sum_{i=0}^{m-1} r_i{}^2 \qquad \ldots.(7)$$

3. Moment Ratio Descriptor

For image block $f(x,y)$, the moment of order $(p+q)$ of a zero-mean block (f), around its center point (x_c,y_c), is defined as:

$$M(p,q)=\sum_{y}\sum_{x}(x-x_c)^p(y-y_c)^q\left[f(x,y)-\bar{f}(x,y)\right] \quad \ldots(8)$$

So, for the domain and range blocks, the first order moments are:

$$M_d(1,0)=\sum_{j=0}^{k-1}\sum_{i=0}^{k-1}(x_i-k_c)\left(d_{ij}-\bar{d}\right) \quad \ldots(9)$$

$$M_d(0,1)=\sum_{j=0}^{k-1}\sum_{i=0}^{k-1}(y_i-k_c)\left(d_{ij}-\bar{d}\right) \quad \ldots(10)$$

$$M_r(1,0)=\sum_{j=0}^{k-1}\sum_{i=0}^{k-1}(x_i-k_c)\left(r_{ij}-\bar{r}\right) \quad \ldots(11)$$

$$M_r(0,1)=\sum_{j=0}^{k-1}\sum_{i=0}^{k-1}(y_i-k_c)\left(r_{ij}-\bar{r}\right) \quad \ldots(12)$$

Where, M_d is the domain moment, M_r is the range moment; (x_i,y_i) are the x and y coordinates of (i,j) pixel, k is the block length and k_c is the value of x and y coordinates of the center point, it is [7]:

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

Now, let us consider the following moments ratio factor (R):

$$R=\frac{M^2(0,1)-M^2(1,0)}{M^2(0,1)+M^2(1,0)} \quad \ldots(14)$$

So, the moment ratio factors for domain and range blocks are:

$$R_d=\frac{M_d^2(0,1)-M_d^2(1,0)}{M_d^2(0,1)+M_d^2(1,0)} \quad \ldots(15)$$

$$R_r=\frac{M_r^2(0,1)-M_r^2(1,0)}{M_r^2(0,1)+M_r^2(1,0)} \quad \ldots(16)$$

It is easy to prove that the magnitude of (R) factor is rotation and reflection invariant. By combining equation (1) with equations (11, 12), and substitute the results in equation (16) it leads to[7]:

$$R_d=R_r \quad \ldots(17)$$

This results implies that "*if the two blocks (range and domain) nearly satisfy the contractive affine transform (i.e., equation 1), then their moment ratio factors (R_d and R_r) should have similar values. This doesn't mean that any two blocks have similar R factor should necessarily satisfy the affine transform*". This fact is utilized to improve (speeding up) the range-domain search task. So, instead of IFS-matching all domain blocks with each range block, only the domain blocks whose (R) magnitudes are similar to that of the tested range block are IFS-matched[7].

4. Symmetry mappings 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 by using a set of isometric transforms (i.e., rotation and flipping), such that eight versions (blocks) are produced for each domain block [8]. 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), (see table (1)).

The main disadvantage of using the isometric mappings in the encoding process is that more

computational time will be required to perform the extra matching processes

5. Encoding Process

To implement this selective search scheme the following block indexing algorithm had been implemented:

1. Load BMP image and put it in (R,G,B) array (three 2D arrays).
2. Convert (R,G,B) array to (Y,U,V) array
3. Down sample the components U and V.
4. For each color component (i.e., the original Y, and the down sampled U,V) do:
   A. Construct the domain and range pools.
   B. For each domain block listed in domain pool:
      a. Determine its moments, $M_d(1,0)$ and $M_d(0,1)$ using equations (9 and 10), then determine its moments ratio (R_d) using equation (15).
      b. Determine the moments ratio index value (I_d) using the following equation:

$$I_d = round\left(\left|R_d\right| xN_{max}\right) \quad \text{.....(18)}$$

      Where, N_{max} is number of R_d-bins, it represents the maximum moment's ratio index value, taking into consideration that the magnitude of (R_d) doesn't exceed (1).
      c. Register the block coordinates ($posI$) of the domain block, and its calculated moments ratio index value (I_d) in a temporary array (L) of records.
   C. Sort the records of the array (L) in ascending order according to their moments index values (I_d).
   D. Establish a set of pointers P to address the start and end of each block of records that have same index (I_d) value.
   E. For each range block listed in the range pool:
      a. Determine its moment ratio (R_r) using equation (16), and the corresponding moment index value (I_r) using the following equation:

$$I_r = round\left(\left|R_r\right| xN_{\max}\right) \quad \text{....(19)}$$

      b. Use the set (p) of pointers and the temporary list $\{L\}$ of records; to match only the domain blocks whose I_d values equal to I_r. In each matching instance determine the scale (s) and error (χ^2), by equations (4 and 5), for the 8 possible isometry cases. Each determined value of (s), should be bounded to be within the predefined extent $[-s_{max}, s_{max}]$.
      c. Compare the result χ^2 of each matching instance with the minimum χ^2_{min} (updated in previous matching instances). If χ^2 is smaller than χ^2_{min} then set $\chi^2_{min} = \chi^2$, and register the associated values of $(s_q, Sym, PosI)$ of the matched domain block as the best IFS matching parameters.
      d. In case that the new registered minimum χ^2_{min} is less than the predefined threshold (E_{min}) then the search process is stopped, and the set $(s_q, Sym, PosI)$ is output as a best encountered IFS match

for the tested range block, then go to step (*4e*).

e. Otherwise, start testing the next domain blocks whose (I_d) values are ($I_r \pm 1$) to get the best IFS match, the minimum threshold value in such case is taken little bit higher than E_{min} (i.e, it is taken E_m). If an acceptable match instance was not met, then the next set of domain blocks whose (I_d) values are ($I_r \pm 2$) should tested. And so forth, till reaching the domain blocks whose moments ratio index is ($I_r \pm W_r$), in such case the registered IFS code associated with minimum error (χ^2_{min}) is considered as the best attained match. The search window [$I_r - W_r$, $I_r + W_r$] represents the extent of IFS matching trials.

Output the set of IFS codes ($i_s, i_r, Sym, PosI$) for the tested range block, and return to step (*4e*).

6. Blocks Isometric State Assignment

Table (2) shows the symmetry mapping of the considered eight transforms. In table (3) the symbol c denotes the coordinates of the center point of the mapped square block (whose size is mxm).

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

$$M_{10} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(x-c) \quad \ldots.(20)$$

$$M_{01} = \sum_{y=0}^{m-1}\sum_{x=0}^{m-1} I(x,y)(y-c) \quad ..(21)$$

By combining both equations (20) and (21) with the equations listed in table (3), the relationship between the new moments values (*M'10*, *M'01*) of the mapped block (using isometric mappings) with its old moments values (*M10*, *M01*) could determined, table (4) lists these relationships.

In this section, a new method for block classification according to its isometric state is described; the classification is based on applying three Boolean criteria, they depend on the status of its first order moments (i.e., M10, M01). These used criteria are:

1. Is $|M_{10}| \geq |M_{01}|$ or not ?
2. Is $|M_{10}| \geq 0$ or not ?
3. Is $|M_{01}| \geq 0$ or not ?

The use of these three Boolean criteria on any block leads to eight block states, as shown in table (5).

Now, if the relationship between the new and old moment values is taken into consideration when the block is mapped by one of the considered isometric mapping (see table 4), then the relationship between the indexes of block could be established (see table 6).

The arrangement of the contents of table (6) could be inverted such that the type of transform needed to map the block from certain isometric state to other state is assigned, see table (7).

6. Test Result

The developed systems have been established using Visual Basic (version 6.0) programming language, and they work under Microsoft windows XP Professional operating system. This set of tests was performed to study the effect of symmetry on the compression

performance parameters of the reconstructed frame and using moment feature to reducing type of symmetry from 8 to 1.

Symmetry Tests

These tests were conducted to investigate the effect of each symmetry operation [0-7]; two kinds of tests have been conducted. the first test full symmetry on compression performance parameters is used as traditional method. The second test is utilized to predict the type of isometric process needed to be applied on the domain block to ensure the best IFS-matching state with the tested range block. In the second test the elapsed time (coding time) will reduce about 8 times. Table (8) shows some compares in time in two cases (predicted process, traditional) in this tests the coding parameters were taken as shown in the table (7). In the second tests were concentrated on studying the effect of the following parameters Frame number, Block length, Jump Step, Minimum Block Error, scale bits, offset bits, Max Scale on the probability distribution of symmetry operation. Figure (1) shows the results of applying these parameters and its effects on the some measures criteria like (MAE, PSNR, Compression Ration and elapsed time). There are large differences between encoding time in both cases (Traditional and predicted).

7. Conclusions

The use of symmetry predictor causes a speed-up in encoding process, and the use of moment features prediction causes more significant speeding-up in encoding process. But, the image quality had little degraded in comparison with its level when the traditional method is applied.

References

[1]Barnsly, M. F., *'' Fractal Everywhere",* Academic Press Inc., 1988.

[2]Jacquin, A. E. *"Image Coding Based on afractal theory of Iterrated Contractive Image Transformation",* IEEE Transaction on Image Processing. 1(1):18-30,1992

[3]Saupe, D.,*" The futility of square isometries in Fractal Image Compression*", IEEE Int. Conf. On Image Processing (ICIP96), Lausanne, Sept. 1996

[4]Schebe, M. "*Square Isometries as Integer Part of Fractal Transformation- An Analysis* ", FREQENZ, 12 Dec.,1996

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

[6]Yokoyama, T., Watanabe, T., and Sugawara, K., *"Similarity-Based Image Retrieval System Using PIFS Codes"*, Graduate School of Information System, University of Electro-Communication, Japan, 2003.

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

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

Table (1) The eight isometric transformations

Sym	Equations	Results
0.Identity	$x' = x\cos(0) + y\sin(0)$: $y' = -x\sin(0) + y\cos(0)$	$x' = x$: $y' = y$
1.Rot.(+90)	$x' = x\cos(90) + y\sin(90)$: $y' = -x\sin(90) + y\cos(90)$	$x' = y$: $y' = -x$
2. Rot.(+180)	$x' = x\cos(180) + y\sin(180)$: $y' = -x\sin(180) + y\cos(180)$	$x' = -x$: $y' = -y$
3.Rot.(+270)	$x' = x\cos(270) + y\sin(270)$: $y' = -x\sin(270) + y\cos(270)$	$x' = -y$: $y' = x$
4.Ref. at x-axis	$x' = -x\cos(0) + y\sin(0)$: $y' = -x\sin(0) + y\cos(0)$	$x' = -x$: $y' = y$
5. Ref. & Rot. (90)	$x' = -x\cos(90) + y\sin(90)$: $y' = -x\sin(90) + y\cos(90)$	$x' = -y$: $y' = -x$
6. Ref. & Rot. (180)	$x' = -x\cos(180) + y\sin(180)$: $y' = -x\sin(180) + y\cos(180)$	$x' = x$: $y' = -y$
7. Ref & Rot. (270)	$x' = -x\cos(270) + y\sin(270)$: $y' = -x\sin(270) + y\cos(270)$	$x' = y$ $y' = x$

Table (2) symmetry mapping

ID	Operation
0	**No operations**
1	**Rotation-90**
2	**Rotation-180**
4	**Rotation-270**
4	**Reflection around Y-axis**
5	**Reflection with rotation-90**
6	**Reflection with rotation-180**
7	**Reflection with rotation-270**

Table (3) The considered isometric mappings

ID	Transform	Mapping Equations
0	**No operation**	***x' = x : y' = y***
1	**Rotation_90**	***x'=(x-c)cos(90)+(y-c)sin(90)+c=y*** ***y'=-(x-c)sin(90)+(y-c)cos(90)+c=2c-x***
2	**Rotation_180**	***x'=(x-c)cos(180)+(y-c)sin(180)+c=2c-x*** ***y'=-(x-c)sin(180)+(y-c)cos(180)+c=2c-y***
3	**Rotation_270**	***x'=(x-c)cos(270)+(y-c)sin(270)+c=2c-y*** ***y'=-(x-c)sin(270)+(y-c)cos(270)+c=x***
4	**Reflection**	***x'=2c-x : y'=y***
5	**Reflection with rotation_90**	***x'=(-x-c)cos(90)+(y-c)sin(90)+c=y*** ***y'=-(-x-c)sin(90)+(y-c)cos(90)+c=x***
6	**Reflection with rotation_180**	***x'=(-x-c)cos(180)+(y-c)sin(180)+c=x*** ***y'=-(-x-c)sin(180)+(y-c)cos(180)+c=2c-y***

7	Reflection with rotation_270	$x'=(-x-c)\cos(270)+(y-c)\sin(270)+c=2c-y$ $y'=-(-x-c)\sin(270)+(y-c)\cos(270)+c=-x$
Where, $c=(m-1)/2$		

Table (4) The relationship between moments before and after the transform

Tran. ID	Transform	Relationship
0	No operation	$M'10= M10$ $M'01= M01$
1	Rotation_90	$M'10=M01$ $M'01=M10$
2	Rotation_180	$M'10= -M10$ $M'01=-M01$
3	Rotation_270	$M'10=-M01$ $M'01=M10$
4	Reflection	$M'10=-M10$ $M'01=-M01$
5	Reflection with rotation_90	$M'10=M01$ $M'01=M10$
6	Reflection with rotation_180	$M'10=M10$ $M'01=-M01$
7	Reflection with rotation_270	$M'10=-M01$ $M'01=-M10$

Table (5) The truth table for the eight blocks states

Block's Class index	Boolean Criteria		
	$\|M_{10}\| \geq \|M_{01}\|$	$\|M_{10}\| \geq 0$	$\|M_{01}\| \geq 0$
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

Table (6) The Block's state indexes before and after isometric Mappings

Old Class Index (Nop)	New Class Index						
	R90	R180	R270	M	M+R90	M+R180	M+R270

0(TTT)	6(FTF)	3(TFF)	5(FFT)	2(TFT)	4(FTT)	1(TTF)	7(FFF)
1(TTF)	4(FTT)	2(TFT)	7(FFF)	3(TFF)	6(FTF)	0(TTT)	5(FFT)
2(TFT)	7(FFF)	1(TTF)	4(FTT)	0(TTT)	5(FFT)	3(TFF)	6(FTF)
3(TFF)	5(FFT)	0(TTT)	6(FTF)	1(TTF)	7(FFF)	2(TFT)	4(FTT)
4(FTT)	1(TTF)	7(FFF)	2(TFT)	5(FFT)	0(TTT)	6(FTF)	3(TFF)
5(FFT)	3(TFF)	6(FTF)	0(TTT)	4(FTT)	2(TFT)	7(FFF)	1(TTF)
6(FTF)	0(TTT)	5(FFT)	3(TFF)	7(FFF)	1(TTF)	4(FTT)	2(TFT)
7(FFF)	2(TFT)	4(FTT)	1(TTF)	6(FTF)	3(TFF)	5(FFT)	0(TTT)

R90≡ Rotation_90; R180≡ Rotation_180; R270≡ Rotation_270; ;M≡ Mirror or Reflection;

Table (7) The required isometric operation to convert the block state

		New blocks state index							
		0	1	2	3	4	5	6	7
Old Block State 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;

Table (8) fixed coding parameters

Fixed(Default)Parameters	values	Fixed(Default)Parameters	values
NoBin(F-level)	200	OfsetBit (Y,U,V)	8
Window Size	1	BitScl (Y,U,V)	6
BlockLen	4	MaxScl (Y,U,V)	3
MinBlkErr(Threshold)	1	MinClsErr	1.5
JmpStp (Y,U,V)	1		

Table (9) some compares between Traditional and prediction methods

				Elapsed Time(sec)	
Tested Parameter \Measures	MAE	PSNR	Comp. Ratio	Suggested Predictor	Traditional
Frame number	6.7	26.8	9.06	99.4	788.8
Block Length	8.7	26	16	157	910.5
Minimum Block error	8.6	26	9	141	813.6
Jump Step	8.9	25.8	9.8	64	650
Offset Bit	10	25.1	9.6	179.8	799.7
Scale Bits	8.7	25.9	9.3	141.4	812.5
Max Scale	8.7	25.9	9.1	126.6	722.8

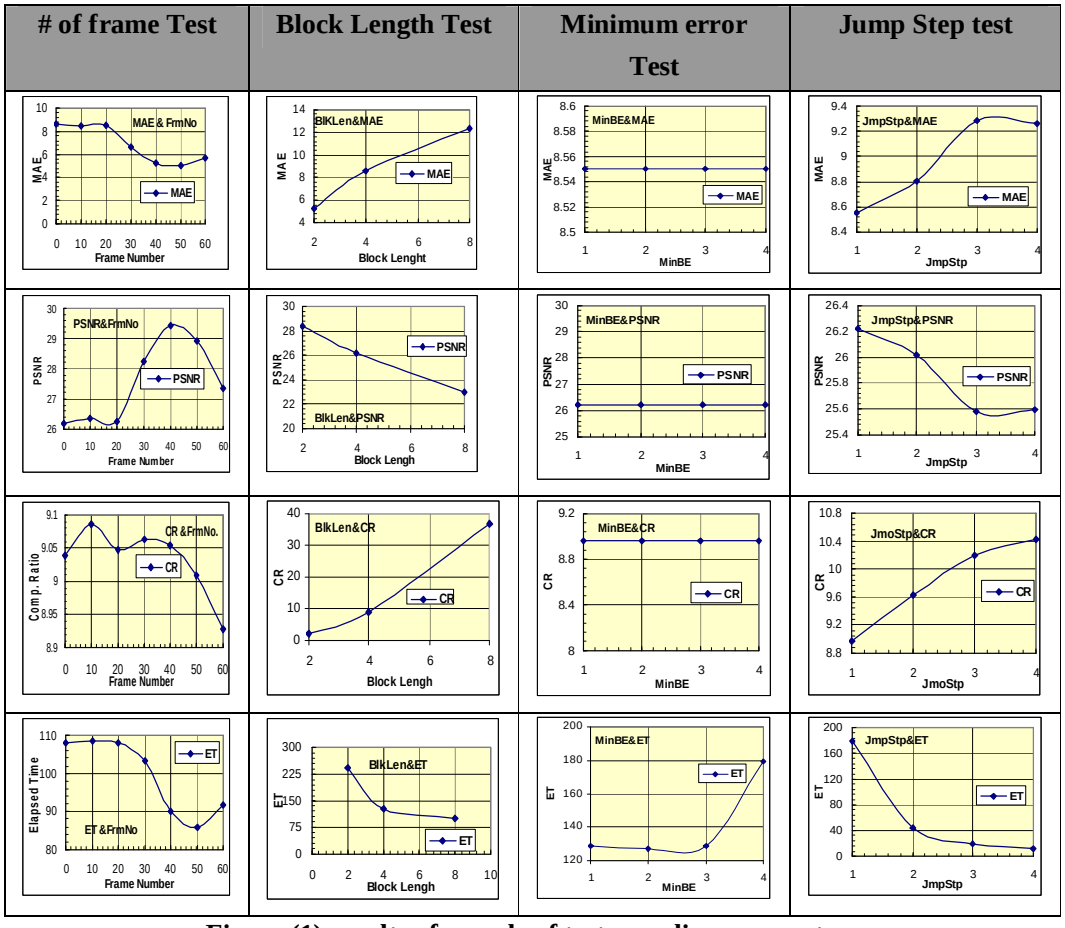

Figure (1) results of sample of test encoding parameters