International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Fractal Color Image Compression Using YP_bP_r and YUV Color Spaces By Zero-Mean Method

Eman A. Al-Hilo[1], Abbas F. Al-Rufaee[2]

[1] Kufa University, Faculty of Education for Girls, Department of Physics, Najaf City, Iraq

[2]Kufa University, Faculty of Science, Department of Physics, Najaf City, Iraq

Abstract: ***In this work , two type of the color space ware used , it is (YP_bP_r , YUV), and Lena color image were used , to test the pest PSNR, CR, ET, by used the fractal image compression by Zero-Mean method. In (YP_bP_r)color space the calculated results were found to be (PSNR=30.99), (CR=10.52), (ET=63.68), while in (YUV) color space is (PSNR=32.75), (CR=10.52), (ET=70.13). So in (PSNR) the (YP_bP_r) model is decreases by (1.76%) than (YUV) model, in CR the (YP_bP_r) model it same in (YUV) model and in ET the (YP_bP_r) model is decreases by (6.45%) than (YUV) model.***

Keywords: YP_bP_r color space, YUV color space, TV color space, Zero-Mean method, Image compression.

1. Introduction

To send and receive any signal, we need to the center of the carrier, may be in the middle of this affects the signal transmitted positively or negatively, the spaces YUV and YP_bP_r are the TV color spaces, YUV is a color space typically used as part of a color image pipeline, YPbPr is the is one of the YUV family of formats such as YUV, YP_bP_r or YC_bC_r [1]. then convert it from RGB to YPbPr using equations 1 and 2, or using equations 3 and 4 to be converted into space YUV, then we work steps listed below[2,3].

$$\begin{bmatrix} Y \\ Pb \\ Pr \end{bmatrix} = \begin{bmatrix} 0.213 & 0.715 & 0.072 \\ -0.115 & -0.385 & 0.500 \\ 0.500 & -0.454 & -0.046 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.575 \\ 1.000 & -0.187 & -0.468 \\ 1.000 & 1.856 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ Pb \\ Pr \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0.000 \end{bmatrix} \left(\begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right) \quad (4)$$

in (2006) studied L. E. George the Zero-Mean equations [4]. While Kasambe and Patel in (2007) studied the coordinate systems used for color image are RGB,YIQ and YUV[5]. in (2013) Eman A. Al-Hilo and Kawther H. Al-khafaji, used Zero-Mean method for matching to show the effect of quantization on the reconstructed image in fractal image compression FIC method [6], and Taha mohammed Hasan and Xingqian Wu in (2013) Published a research paper an Adaptive Fractal Image Compression (AFIC) algorithm is proposed to reduce the long time of the Fractal Image Compression (FIC), for this purpose Zero- Mean Intensity Level Fractal Image Compression based on Quadtree partitioning [7]. Rusul Talib Zehwar in (2014) study the different television color spaces by the traditional method, by using two color image and compare results.[8]

2. IFS Coding for Zero-Mean Blocks

The offset factor is determined using traditional affain mapping, described by equation (5), has wide dynamic range (i.e., [-255,510]), this may cause large errors in some image regions (or points) especially those which belong to high contrast area. The results of the analysis conducted in this research indicate that the traditional offset factors require an additional bit (i.e., sign-bit). Also, the analysis results indicate that the correlation between the offset coefficients of the adjacent blocks is weak and not similar to that found between the mean values of the adjacent blocks. So, to get over this disadvantage a change in IFS mapping equation was performed. For a range block with pixel values (r_o, r_1,…,r_{n-1}), and the domain block (d_o,d_1,…,d_{n-1}) the contractive affine approximation is [4]:

$$r_i = sd_i + o \quad (5)$$

Where, r_i is the optimally approximated ith 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. Taking the average of both sides in equation (5) the following equation is obtained:

$$\bar{r} = s\bar{d} + o \quad (6)$$

where,

$$\bar{r} = \frac{1}{m}\sum_{i=0}^{m-1} r_i \quad (7)$$

$$\bar{d} = \frac{1}{m}\sum_{i=0}^{m-1} d_i \quad (8)$$

The subtraction of equation (6) from equation (5) leads to:

$$\acute{r}_i - \bar{r} = sd_i - s\bar{d} \quad (9)$$

So, this contractive affine transform could be rewritten to become in the form:

$$\acute{r}_i = s(d_i - \bar{d}) + \bar{r} \quad (10)$$

From equation (10), the fractal parameters become (s) and ($\bar{r}$) instead of the conventional (s) and (o) coefficients in traditional IFS mapping equation. The scale (s) parameter could be determined by applying the least mean square difference (ϵ^2) between the approximated ($\acute{r}_i$) and actual (r_i) values [4]:

$$\epsilon^2 = \frac{1}{m}\sum_{i=0}^{m-1}(\acute{r}_i - r_i)^2 \qquad (11)$$

$$\frac{\partial \epsilon^2}{\partial s} = 0 \qquad (12)$$

The straight forward manipulation for equations (11), (12) leads to:

$$s = \begin{cases} \dfrac{\frac{1}{m}\sum_{i=0}^{m-1} d_i r_i - \bar{r}\bar{d}}{\sigma_d^2} & if\ \sigma_d^2 > 0 \\ 0\ if\ \sigma_d^2 = 0 \end{cases} \qquad (13)$$

$$\epsilon^2 = \sigma_r^2 + s\left[s\sigma_d^2 + 2\bar{d}\bar{r} - \frac{2}{m}\sum_{i=0}^{m-1} d_i r_i\right] \quad (14)$$

where, $\sigma_d^2 = \frac{1}{m}\sum_{i=0}^{m-1} d_i^2 - \bar{d}^2$ (15) $\sigma_r^2 = \frac{1}{m}\sum_{i=0}^{m-1} r_i^2$ (16)

3. Encoding in the Zero-Mean Method Process

Encoding in the zero-mean method could be summarized by the following figure(1)[1,9]:

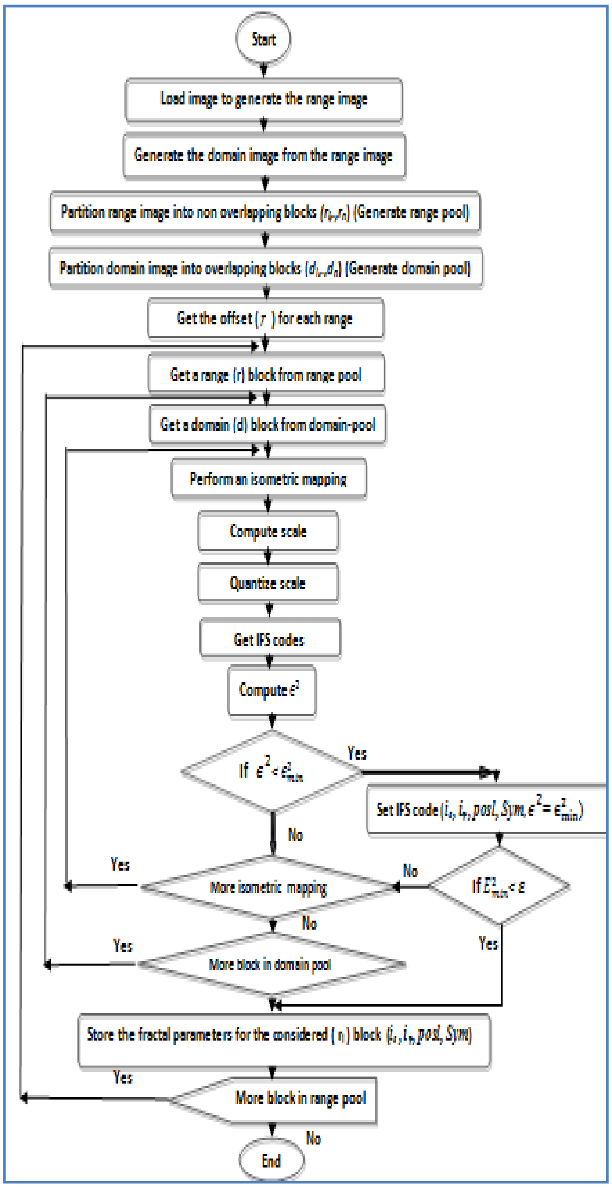


Figure 1: Flowchart of the Zero-Mean encoding

4. Decoding in the Zero-Mean Method Process

The decoding process by zero-mean method can be summarized in the following figure(2)[1]:

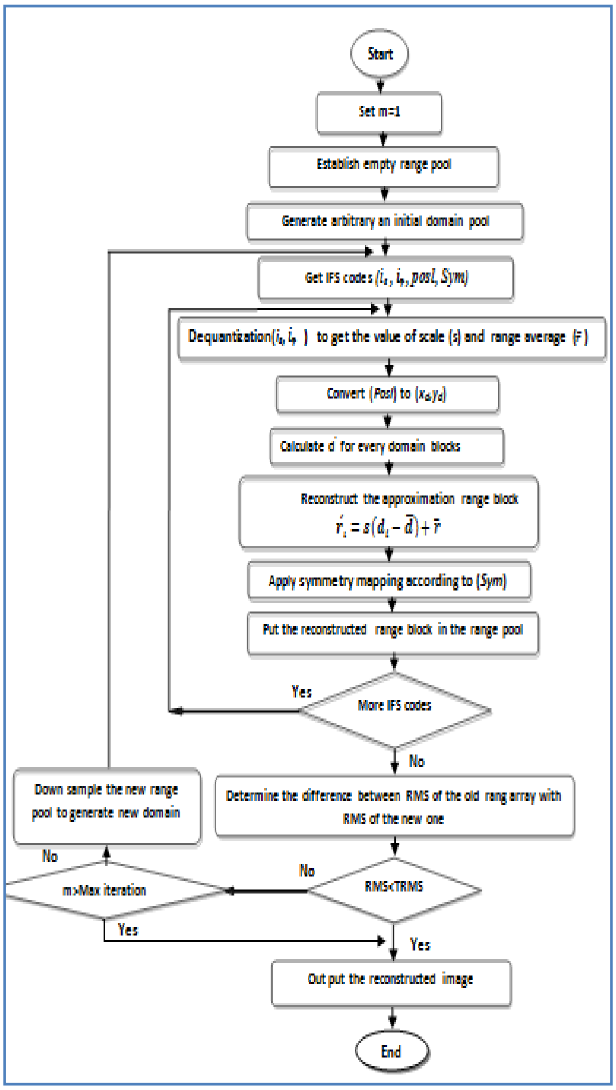


Figure 2: Flowchart of the Zero-Mean decoding

5. Tests Results

To evaluate the performance of the established color FIC system by Zero-Mean method, several tests were conducted and the results of these tests. In all these tests Lena image (256x256, 24 bit per pixel) was used as a testing object. The values of the parameters Min Offset and Max Offset were fixed in all these tests at (0) and (255) respectively and TMSE is 6 . To test the effect of each parameter the values of other parameters were fixed for two color spaces. All the computations conducted in this research work have been accomplished by utilizing some dedicated programs; these programs have been designed and implemented to achieve the coding and testing tasks. Visual Basic (Version 6) was used as programs development tool to construct the required programs. The environment used of our tests is a Laptop (DELL) : intel (R) core (TM) 2Due CPU T6600 @ 2.20GHZ Processor, 32-bit Operating System and 2.00 GB RAM.

5.1 Maximum and Minimum Scale Tests

This set of tests was applied to study the effect of Min Scale and Max Scale. Table 1 is show the result this test and figures 3,4,5 show the graphic for this results .

Table 1: Maximum and Minimum Scale Tests

Max Scale	Min Scale	YP_bP_r PSNR	YP_bP_r CR	YP_bP_r ET (sec)	YUV PSNR	YUV CR	YUV ET (sec)
0.5	-0.5	29.69	10.52	61.02	30.51	10.52	63.61
	-1	30.92	10.52	61.79	32.44	10.52	64.37
	-1.5	30.87	10.52	62.45	32.44	10.52	65.02
	-2	30.79	10.52	62.18	33.31	10.52	64.68
1	-0.5	30.89	10.52	61.98	32.27	10.52	64.44
	-1	30.99	10.52	63.68	32.64	10.52	67.24
	-1.5	30.92	10.52	65.22	32.55	10.52	68.29
	-2	30.88	10.52	65.21	32.54	10.52	68.07
1.5	-0.5	30.89	10.52	62.68	32.25	10.52	65.04
	-1	30.98	10.52	65.37	32.59	10.52	68.08
	-1.5	30.76	10.52	67.33	32.31	10.52	69.71
	-2	30.68	10.52	67.48	32.21	10.52	69.75

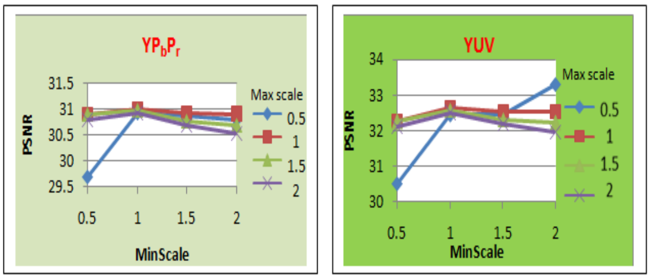


Figure 3: effect of Min scale on PSNR for two color spaces

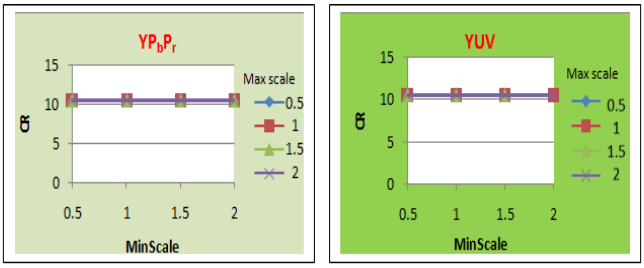


Figure 4: effect of Min scale on CR for two color spaces

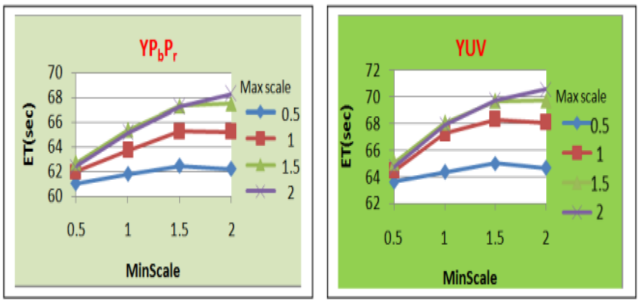


Figure 5: effect of Min scale on ET for two color spaces

5.2 Domain Size Tests

This set of tests was performed to study the effects of DomSize parameter on the compression performance. Table 2 is show the result this test and figures 6,7,8 show the graphic for this results .

Table 2: Domain Size Tests

Original image	Dom Size 128x128	Dom Size 64x64	Dom Size 32x32	Dom Size 16x16
				
PSNR YP_bP_r	30.99	30.00	28.69	26.66
CR YP_bP_r	10.52	10.52	10.52	10.52
ET YP_bP_r	63.68	17.37	04.71	01.61
				
PSNR YUV	32.64	31.23	29.63	27.33
CR YUV	10.52	10.52	10.52	10.52
ET YUV	67.24	18.05	04.81	01.64

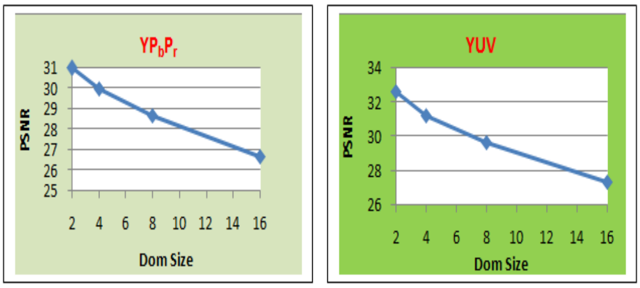


Figure 6: effect of Dom Size on PSNR for two color spaces

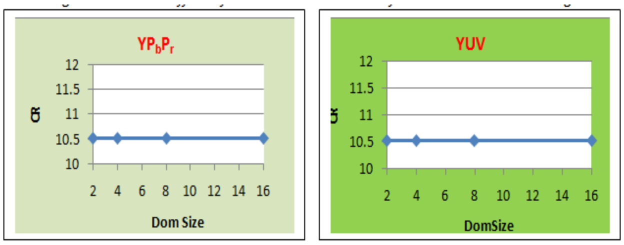


Figure 7: effect of Dom Size on CR for two color spaces

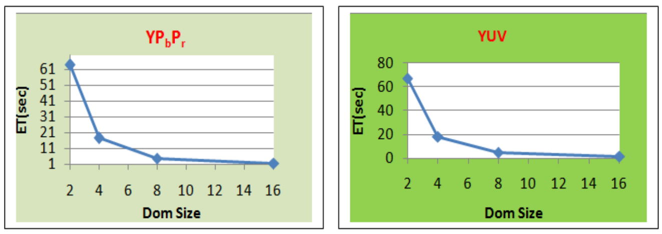


Figure 8: effect of Dom Size on ET for two color spaces

5.3 Block Size Tests

Table 3 show the result this test and figures 9,10,11 show the graphic for this results .

Table 3: Block Size Tests

Original image	BlockSize (2x2)	BlockSize (4x4)	BlockSize (8x8)	BlockSize (16x16)
				
PSNR YP_bP_r	34.59	30.99	26.53	22.87
CR YP_bP_r	02.43	10.52	45.85	201.4
ET YP_bP_r	26.48	63.68	52.96	42.57
				
PSNR YUV	40.47	32.64	27.07	23.22
CR YUV	02.43	10.52	45.85	201.4
ET YUV	29.66	67.24	53.25	42.63

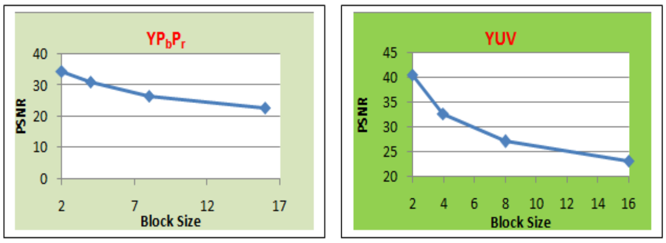


Figure 9: effect of Block Size on PSNR for two color spaces

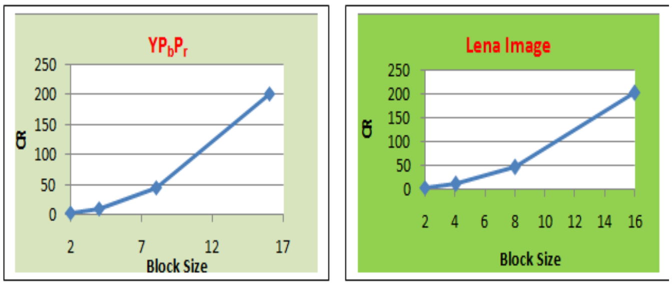


Figure 10: effect of Block Size on CR for two color spaces

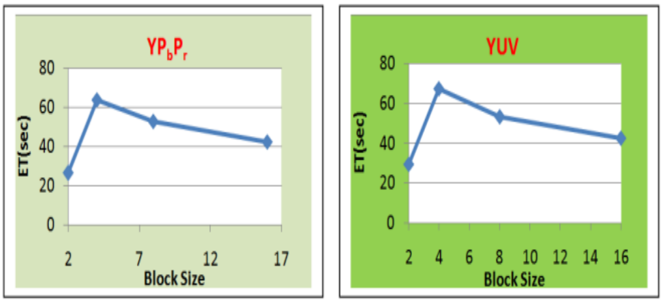
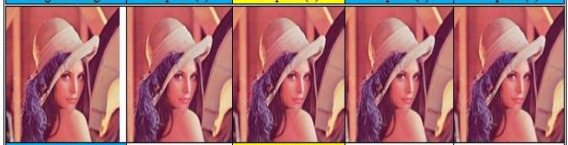


Figure 11: effect of Block Size on ET for two color spaces

5.4 Step Size Tests

Table 4 is show the result this test and figures 12,13,14 show the graphic for this results .

Table 4: Step Size Tests

Original image	StepSize(1)	StepSize(2)	StepSize(3)	StepSize(4)
				
PSNR YP_bP_r	31.41	30.99	30.50	30.33
CR YP_bP_r	10.52	10.52	10.52	10.52
ET YP_bP_r	240.8	63.68	29.64	17.21
				
PSNR YUV	32.25	32.64	31.94	31.53
CR YUV	10.52	10.52	10.52	10.52
ET YUV	252.7	67.24	31.18	18.22

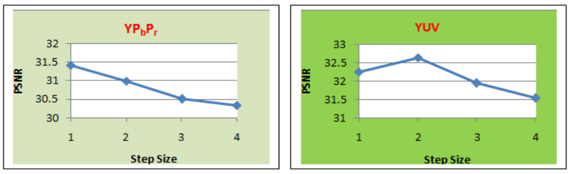


Figure 12: effect of Step Size on PSNR for two color spaces

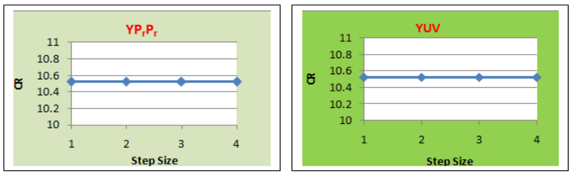


Figure 13: effect of Step Size on CR for two color spaces

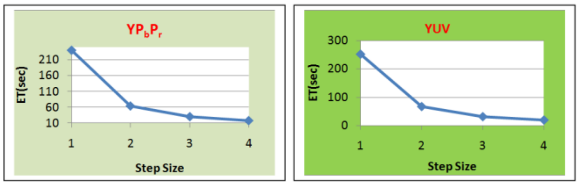


Figure 14: effect of Step Size on ET for two color spaces

5.5 Permissible Error Levels ($\mathcal{E}_o$) Tests

Table 5 is show the result this test and figures 15,16,17 show the graphic for this results .

Table 5: Permissible Error Levels ($\mathcal{E}_o$) Tests

$\mathcal{E}_o$	YP_bP_r			YUV		
	PSNR	CR	ET (sec)	PSNR	CR	ET (sec)
0.1	30.99	10.52	74.01	32.64	10.52	73.89
0.2	30.99	10.52	74.05	32.64	10.52	73.86
0.3	30.99	10.52	73.77	32.64	10.52	73.50
0.4	30.99	10.52	74.64	32.64	10.52	73.03
0.5	31.00	10.52	72.25	32.64	10.52	72.26
0.6	30.99	10.52	70.29	32.64	10.52	70.04
0.7	31.00	10.52	67.41	32.64	10.52	67.15
0.8	30.99	10.52	63.68	32.62	10.52	62.74
0.9	30.98	10.52	60.90	32.61	10.52	59.65
1	30.96	10.52	57.85	32.59	10.52	56.73
2	30.64	10.52	36.21	32.12	10.52	35.19

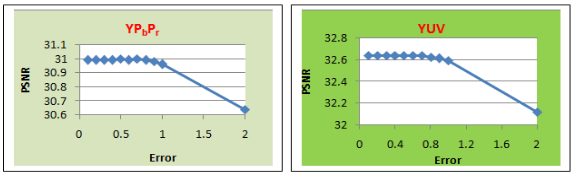


Figure 15: effect of (Ɛo) on PSNR for two color spaces

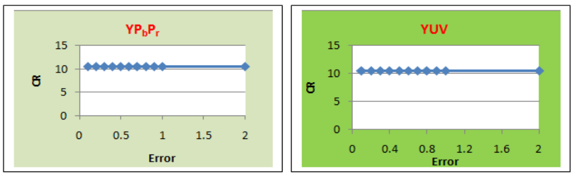


Figure 16: effect of (Ɛo) on CR for two color spaces

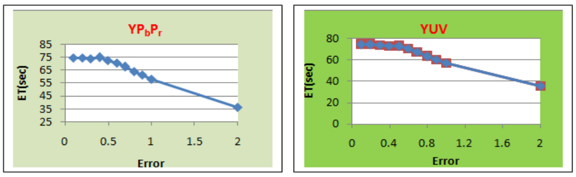


Figure 17: effect of (Ɛo) on ET for two color spaces

6. Conclusion

- The case of Min Scale (-1) and Max Scale (1),leads to high PSNR, for tow color spaces.
- The Domain Size (128x128) leads to best PSNR, for tow color spaces.
- The Block Size (4x4) leads to best PSNR, for tow color spaces.
- The Step Size (2) leads to best PSNR, for tow color spaces.
- The Permissible Error Levels (ε_o) (0.8) leads to best PSNR, for YP_bP_r color space and (0.7) for YUV color space.
- The PSNR parameter in the (YP_bP_r) model is decreases by (1.76%) than (YUV) model.
- The CR parameter in the (YP_bP_r) model it same in (YUV) model.
- The ET parameter in the (YP_bP_r) model is decreases by (6.45%) than (YUV) model.
- From the results above the color space (YUV) is better than (YP_bP_r).

References

[1] Mikael Andreen , "Camera ISP applied on a programmable GPU" , Lund University , 2008.
[2] Al-Hilo, E.A., "Speeding-up Fractal Colored Image Compression using Moments Features ", PhD Thesis, College of Science, AL-Mustansiriyah University, Baghdad, 2007.
[3] jaaskelainen, Joni, "TV-Out Image Sharpness Characterization Using Output Signal Measurements ", Master's Thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, 2009.
[4] George, L., "IFS Coding for Zero-Mean Image Blocks", Iraqi Journal of Science, Vol.47, No.1, 2006.
[5] Kasambe, P., and Patel, M., "Modified Fractal Image Compression Using Genetic Algorithms", Proceeding of SPIT- IEEE Colloquium and International Conference, Mumbai, India, Vol. 1, 134, 2007.
[6] Al-Hilo, E. A., Al-khafaji, K. H., " Effect Of Quantization Process On FIC For Gray Image by Using Zero-Mean Method", Journal of Kufa – Physics Vol.5/ No.2, PP 8-14 , 2013.
[7] Hasan, Taha mohammed, Wu, Xingqian," An Adaptive Fractal Image Compression ", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, PP 98-108, March 2013 .
[8] Zehwar, Rusul T., " Comparison of Fractal Color Image Compression Using Different Color Space ", a thesis for master, University of Kufa, Iraq, 2014.