

# Comparison Fractal Color Image Compression using YIQ and YUV Color Model

Eman A. Al-Hilo

Physics Department/ College of Education for Girls  
Kufa University/ Iraq

Rusul Zehwar

Physics Department/ College of Education for Girls  
Kufa University/ Iraq

**Abstract**—The principle of fractal image coding is that the image converges to a stable image by iterating the contractive transformations on an arbitrary initial image. This algorithm Partition the image into a number of range blocks and domain blocks. For every range block, the best matching domain block is searched for among all domain blocks by performing a set of transformations on each block. For color image compression, the Fractal coding is applied on different planes of color image independently by treating each plane as gray level image. The coordinate systems used for color image are RGB, YIQ and YUV. To encode a color image the main idea is to divide the image into its three different layers or components (RGB, YIQ and YUV). It is then possible to compress each of these layers separately, handle each of the layers as an independent image. In this paper the data of the color component (R,G,B) are transformed two times in two program separately, ones for YIQ and other for YUV color space. The results show that using (YUV) color space is more useful and efficient than using YIQ in fractal image compression, where PSNR increase 0.1% , CR increase 0.31% and ET decrease 2.321%.

**Keywords**—Compression image; fractal color image compression and iterated function system

## I. INTRODUCTION

Colors are important for human for communicating with the daily encountered objects as well as his species, these colors should be represented formally and numerically within a mathematical formula so it can be projected on device computer storage and applications, this mathematical representation is known as color model that can hold the color space, by the means of color's primary components (Red, Green, and Blue) the computer can visualizes what the human does in hue and lightness. Most of these techniques reduce the redundancies between color components by transforming the color primaries into a decorrelated color model such as YUV and YIQ. [1]

The YIQ color space has been used in NTSC (National Television System Committee) color TV system and has been employed in USA, Canada, Japan and Korea. The Y stands for luminance components, and I and Q stand for chrominance information for representing color. [2]

RGB values can be transformed to YIQ color space using equation (1), while YIQ values can be converted back to RGB using equation (2). [3,4]

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.247 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots\dots\dots(1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.105 & 1.702 \end{bmatrix} \cdot \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \dots\dots\dots(2)$$

The YUV color space defines is a term of one luminance (Y channel) and two chrominance components (UV channels).U represents the color difference between blue signal and luminance (B–Y) and V represents the difference between red and luminance (R–Y). RGB values can be transformed to YUV color space using equation (3), while YUV values can be converted back to RGB using equation (4). [3,4]

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots\dots\dots(3)$$

This is the inverse matrix to get the RGB components out of the YUV color:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \dots\dots\dots(4)$$

Many researches study the color image compression. Porat [2001] study the effect of intercolor correlation on color image compression. He has been compare color compression techniques that take advantage of this inter color correlation. Most of these techniques reduce the redundancies between color components by transforming the color primaries into a de correlated color space, such as YIQ or YUV. [5]

Kasambe and Patel [2007] they have to provide a method that uses genetic algorithms to speed up. The coordinate systems used for color image are RGB, YIQ and YUV. The time required for the account in fractal image compression with acceptable Image quality for color and gray scale images. [2]

Al-Hilo [2007] has studied speeding fractal color image compression by moment feature includes converting the RGB model system to YUV model and minimize of the rang of U and V because of most of the image data are concentrated in the range of Y [6]

## II. MATCHING PROCESS

After the generation of the range and domain pools; one takes each range block listed in the range pool and map it with all the domain blocks listed in the domain pool. At each mapping instance one determines the mapping coefficients, i.e., scale (s) and offset (o), which are called the IFS coefficient. These parameters (s) and (o) are determined by applying the least sum  $X^2$  of square errors between  $r_i'$  and  $r_i$  according to following equation: [7]

$$x^2 = \sum_{i=0}^{n-1} (r_i' - r_i)^2 \dots\dots\dots(5)$$

The minimum of  $x^2$  occurs when:

$$\frac{\partial x^2}{\partial y} = 0 \text{ and } \frac{\partial x^2}{\partial o} = 0 \dots\dots\dots(6)$$

Substituting equation (5) in (6) and using equations (7) one gets:

$$S = \frac{n \sum_{i=0}^{n-1} r_i d_i - \sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} d_i}{n \sum_{i=0}^{n-1} d_i^2 - (\sum_{i=0}^{n-1} d_i)^2} \dots\dots\dots(7)$$

$$O = \frac{\sum_{i=0}^{n-1} r_i \sum_{i=0}^{n-1} d_i^2 - \sum_{i=0}^{n-1} r_i d_i \sum_{i=0}^{n-1} d_i}{n \sum_{i=0}^{n-1} d_i^2 - (\sum_{i=0}^{n-1} d_i)^2} \dots\dots\dots(8)$$

$$x^2 = \frac{1}{n} [\sum_{i=0}^{n-1} r_i^2 + s \sum_{i=0}^{n-1} d_i^2 - 2 \sum_{i=0}^{n-1} r_i d_i + 2o \sum_{i=0}^{n-1} d_i + o(no - 2 \sum_{i=0}^{n-1} r_i)] \dots\dots\dots(9)$$

where,  $d_i$  is the  $i^{\text{th}}$  pixel value of the matched domain block.

$r_i$  is the  $i^{\text{th}}$  pixel value of the range block.

n is the number of pixels in each block (i.e. the block size).

## III. ENCODING TECHNIQUE

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

- 1) Load BMP image and put it in (R,G,B) arrays (i.e., three 2D arrays).
- 2) Convert (R,G,B) arrays to (Y,I,Q) arrays in first program and convert to (Y,U,V) in second program.
- 3) Down sample (I and Q) to quarter of its original size in first program and to (U and V) in second program.
- 4) For each component (i.e., the original Y, and the down sampled (I,Q) for first program and (the original Y, and the down sampled (U,V) for second program) do:
  - a) Establish the range array
  - b) Down sample the range image to produce the domain array.
  - c) Partitioning:
    - (1) The range array must be partitioned into non-overlapping fixed blocks, to generate the range blocks ( $r_1, \dots, r_n$ ).

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

d) Searching:

- (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 (7-8).
- (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 (7-8).
- (6) Compute the approximation error ( $\chi^2$ ) using equation (9).
- (7) After the computation of IFS code and the sum of error ( $\chi^2$ ) of the matching between the range and the tested domain block., the ( $\chi^2$ ) is compared with registered minimum error ( $\chi^2_{min}$ ); such that:  
If  $\chi^2 < (\chi^2_{min})$  then

$$s_{opt} = i_s; o_{opt} = i_o, \chi^2_{min} = \chi^2$$

PosI=domain block index

Sym=symmetry index

End if

- (8) If  $\chi^2_{min} < \epsilon$  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 symmetry 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.,  $i_s, i_o, posI, Sym$ ) 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.

IV. DECODING TECHNIQUE

The decoding process by YIQ and YUV model can be summarized in the following steps:-

1) *Generating 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. This step is called the dequantization step.*

3) *Choosing the number of possible iterations, and the threshold value of the mean square error (TMSE). At each iteration the following steps are performed:*

a) *For each range block one determines 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.:*

$$r_i' \approx sd_i + o \dots \dots \dots (10)$$

c) *The generated block is transformed (rotated, reflected, or both) according to its corresponding IFS symmetry parameter value (Sym).*

d) *The generated  $r_i'$  block is placed in its position in the decoded image array (range image).*

e) *Checking whether there is another range block, if yes then steps (b,c,d) are repeated.*

f) *Down sampling the reconstructed image (range pool) in order to produce the domain pool by using the averaging (or integer) sampling.*

g) *Calculating 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 continued 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 continue till reaching the predefined maximum number of iterations (m), in our program used (m=20).*

4) *The above (steps 3a-3g) should be implemented upon the three components (Y,I,Q in first program and Y,U,V to second program) to produce the attractor of each component.*

5) *Converting the reconstructed (YIQ color components in first program and YUV in second program) to RGB components sing the inverse (YIQ and YUV) transform equations respectively (2,4).*

6) *Calculating the fidelity criteria for each RGB component, then determines the overall value of the fidelity criteria for the RGB reconstructed image.*

V. TESTS RESULTS

This work is carried out in Visual Basic 6.0 version on Laptop (hp): intel (R) Core (TM) i5-2430M CUP @ 2.40 GHz Processor, 64-bit Operating System and 6.00 GB RAM. To evaluate the performance of the established colour FIC system by YIQ and YUV model, the proposed system has been tested using Lena image (256x256 pixel, 24bits) as test image. These tests explore the effect of the following coding parameters on the compression performance parameters of the established system of YIQ color space and YUV color space:

A. Maximum and Minimum Scale Tests

This set of tests was conducted to study the effect of MinScale, and MaxScale on the compression performance parameters of the reconstructed image in YIQ and YUV models. Table (1) and (2) show the effects of this test on compression performance parameters for YIQ and YUV respectively.

TABLE I. EFFECTS OF MAXSCALE AND MINSKALE ON THE COMPRESSION PERFORMANCE PARAMETERS USING (YIQ) COLOR MODEL

MaxScale	MinScale	PSNR	CR	ET(sec)
1	-1	30.85	8.73	54.65
	-1.5	30.94	8.73	54.34
	-2	30.94	8.73	54.26
	-2.5	30.92	8.73	54.46
	-3	30.90	8.73	53.96
2	-1	31.02	8.73	54.99
	-1.5	31.05	8.73	54.17
	-2	31.03	8.73	54.36
	-2.5	31.02	8.73	53.99
	-3	31.01	8.73	54.97
3	-1	31.04	8.73	54.25
	-1.5	31.04	8.73	54.19
	-2	31.05	8.73	54.76
	-2.5	31.04	8.73	56.56
	-3	30.98	8.73	55.27

TABLE II. EFFECTS OF MAXSCALE AND MINSKALE ON THE COMPRESSION PERFORMANCE PARAMETERS USING (YUV) COLOR MODEL

MaxScale	MinScale	PSNR	CR	ET(sec)
1	-1	30.93	9.04	58.59
	-1.5	31.02	9.04	57.98
	-2	31.02	9.04	57.72
	-2.5	31.02	9.04	58.33
	-3	31.02	9.04	57.52
2	-1	31.11	9.04	57.75
	-1.5	31.14	9.04	58.10
	-2	31.15	9.04	57.50
	-2.5	31.15	9.04	57.87
	-3	31.12	9.04	57.91
3	-1	31.11	9.04	58.83
	-1.5	31.13	9.04	58.41
	-2	31.14	9.04	57.64
	-2.5	31.13	9.04	55.33
	-3	31.13	9.04	58.33

1) Step Size Tests

In this set of tests the effect of the shift StepSize parameter is studied. Figures (1) and (2) show the effects of this test using YIQ and YUV models respectively.

Original image	StepSize (1)	StepSize (2)	StepSize (3)	StepSize (4)
				
<b>PSNR</b>	31.5	31.05	30.71	30.48
<b>CR</b>	8.73	8.73	8.73	8.73
<b>ET</b>	208.23	53.45	24.08	14.20

Fig. 1. Effect of StepSize parameter on the compression performance parameters using (YIQ)




Original image	StepSize (1)	StepSize (2)	StepSize (3)	StepSize (4)
				
<b>PSNR</b>	31.60	31.15	30.83	30.57
<b>CR</b>	9.04	9.04	9.04	9.04
<b>ET</b>	223.88	54.68	24.69	14.46

Fig. 2. Effect of StepSize parameter on the compression performance parameters using (YUV)

### 2) Block Size Tests

This set of conducted tests aimed to investigate the effects of the BlockSize parameter on the compression performance parameters. Figures (3) and (4) show the effects of this test using YIQ and YUV models respectively.

Original image	BlockSize (2x2)	BlockSize (4x4)	BlockSize (8x8)	BlockSize(16x16)
				
<b>PSNR</b>	34.32	31.05	27.15	23.68
<b>CR</b>	2.04	8.73	37.46	161.68
<b>ET</b>	30.68	53.57	31.27	24.17

Fig. 3. Effects of BlockSize parameter on the compression performance parameters using (YIQ)



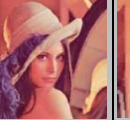
Original image	BlockSize (2x2)	BlockSize (4x4)	BlockSize (8x8)	BlockSize(16x16)
				
<b>PSNR</b>	28.80	31.15	27.19	23.68
<b>CR</b>	2.11	9.04	38.89	168.32
<b>ET</b>	101.70	57.62	31.34	23.59

Fig. 4. Effects of BlockSize parameter on the compression performance parameters using (YUV)

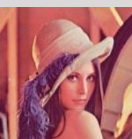

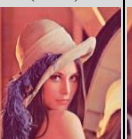
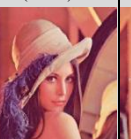
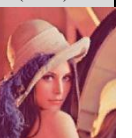
Original image	DomSize (128x128)	DomSize (64x64)	DomSize (32x32)	DomSize (16x16)
				
<b>PSNR</b>	31.05	30.49	29.49	27.80
<b>CR</b>	8.73	8.73	8.73	8.73
<b>ET</b>	53.75	13.24	3.48	1.14

Fig. 5. The effects of DomSize on the compression performance parameters using YIQ

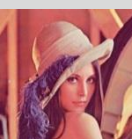
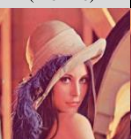
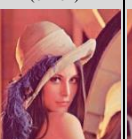
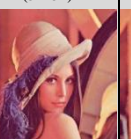
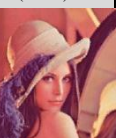
Original image	DomSize (128x128)	DomSize (64x64)	DomSize (32x32)	DomSize (16x16)
				
<b>PSNR</b>	31.15	30.55	29.95	27.85
<b>CR</b>	9.04	9.04	9.04	9.04
<b>ET</b>	54.85	13.68	4.56	1.23

Fig. 6. The effects of DomSize on the compression performance parameters using YUV

### 3) Domain Size Tests

This set of tests was performed to define the effect of the DomSize on the compression parameters. Figures (5) and (6) show the effects of this test using YIQ and YUV models respectively

### 4) Permissible Error Value ( $\epsilon_o$ ) Tests

This set of tests was performed to study the effects of permissible error level ( $\epsilon_o$ ) on the compression performance parameters. Tables (3) and (4) show the effects of this test using YIQ and YUV models respectively

TABLE III. EFFECTS OF PERMISSIBLE ERROR VALUE (E) ON THE COMPRESSION PERFORMANCE PARAMETERS USING (YIQ)

$\epsilon_o$	PSNR	CR	ET(sec)
0.1	31.048	8.73	54.72
0.2	31.048	8.73	55.36
0.3	31.048	8.73	54.49
0.4	31.047	8.73	54.27
0.5	31.048	8.73	54.27
0.6	31.047	8.73	53.62
0.7	31.045	8.73	55.33
0.8	31.044	8.73	55.00
0.9	31.041	8.73	51.67
1	31.033	8.73	50.32
2	30.826	8.73	33.21
3	30.367	8.73	20
4	29.959	8.73	14.22
5	29.534	8.73	10.10

TABLE IV. SHOWS THE DIFFERENT BETWEEN YUV, YIQ MODEL FOR LENA IMAGE

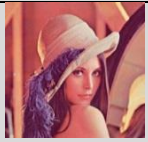
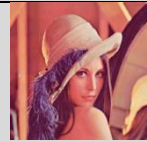
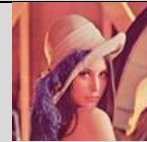


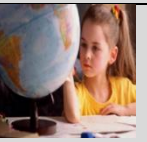
Original image	YIQ	YUV	
			<b>Percentage Error %</b>
<b>PSNR</b>	31.05	31.15	
<b>CR</b>	8.73	9.04	Increase 0.31%
<b>ET</b>	53.75	57.62	increase 2.23%

TABLE V. SHOWS THE DIFFERENT BETWEEN YUV, YIQ MODEL FOR SUN CHILD IMAGE

Original image	YIQ	YUV	
			<b>Percentage Error %</b>
<b>PSNR</b>	35.2	36.32	
<b>CR</b>	8.73	9.04	Increase 0.31%
<b>ET</b>	51.48	55.14	Increase 3.66%

## VI. CONCLUSIONS

The results indicate that using YUV model is more efficient than YIQ model as in following:

- 1) The PSNR is increase 0.10% in YUV model than in (YIQ) model.
- 2) The CR is increase 0.31% than in YUV model than in (YIQ) model.
- 3) The ET parameter in the (YUV) is increase 2.23% than in (YIQ) model.

Table (5) and (6) summarizes the results for Lena and sun child image respectively.

TABLE VI. EFFECTS OF PERMISSIBLE ERROR VALUE (E) ON THE COMPRESSION PERFORMANCE PARAMETERS USING (YUV)

$\epsilon_0$	PSNR	CR	ET(sec)
0.1	31.1466	9.04	53.68
0.2	31.1465	9.04	54.95
0.3	31.145	9.04	54.41
0.4	31.144	9.04	54.26
0.5	31.140	9.04	56.67
0.6	31.142	9.04	55.59
0.7	31.140	9.04	54.54
0.8	31.136	9.04	57.74
0.9	31.133	9.04	51.00
1	31.120	9.04	50.72
2	30.862	9.04	33.42
3	30.151	9.04	20.19
4	29.540	9.04	14.19
5	28.994	9.04	5.05

## REFERENCES

- [1] Ibraheem, N., Hasan, M., Khan, R., Mishra, P., " *Understanding Color Models: A Review*", ARPN Journal of Science and Technology, VOL. 2, NO. 3, April 2012.
- [2] Kasambe p.v. and Patel M., " *Modified Fractal Image Compression Using Genetic Algorithms*", Mumbai, India, Vol. 1, 134, 2007.
- [3] Ford A., Roberts A., " *Color Space Conversions*", August11, 1998(b).
- [4] González J. M. C., Rodríguez M. A. V., Pulido J. A. G., and Pérez J. M. S., " *Digital Signal Processing*", Available online 23 October 2009.
- [5] Porat M., " *The Effect of Inter Color Correlation on Color Image Compression*", M.Sc., Electrical Engineering, January 2001.
- [6] Al-Hilo, E.A., " *Speeding-up Fractal Colored Image Compression using Moments Features*", PhD Thesis, College of Science, AL-Mustansiriyah University, Baghdad, 2007.
- [7] Fisher, Y., " *Fractal Image Compression Theory and Application*", Book, University of California, Institute for Nonlinear Science, Springer-Verlay, New York, Inc, 1995.