

Evaluation of Compressing Haar Wavelet transformed images With Fast Fractal Image Compression

Nevart A. Yousif

Computer Systems Department , Technical Institute - Kirkuk , Iraq

(Received: 27 / 2 / 2012 ---- Accepted: 16 / 9 / 2012)

Abstract

This paper proposed some methods for applying fast fractal image compression (FFIC) on haar wavelet transformed images. The received red, green and blue (RGB) color image is first converted to YC_bC_r color space, Then Haar wavelet transform is applied to each of the subbands Y, C_b and C_r separately. This produces four smaller filtered images or subbands: LL, HL, LH, and HH. Three m are conducted to test the effect of applying fractal compression on these four subbands. In each method the subbands are treated in different way by applying the FFIC on some parts and leave the others without any changes to find the best compression method. The FFIC is speeded up by using the centralized moment descriptors which are applied on each range and domain block, then sort the domain blocks to determine the suitable symmetry case without trying the eight symmetry cases when searching for the best match in the domain blocks. The subbands (HL, LH, and HH) in C_b and C_r components are not saved at all to increase the compression because these parts do not contain important information that affects the quality of the image while the LL part and all Y component parts are managed in different way in each of the three suggested methods. Quantization is applied to reduce the saved data. Finally the approach is tested on Lena's images using the PSNR to test the quality, compression ratio and the compression time parameters.

Keywords : Haar Wavelet, Image compression, Fractal Image Compression, color spaces.

1. Introduction

The Haar transform is the simplest wavelet transform, but even this simple method illustrates the power of the wavelet transform. It turns out that the low levels of the discrete wavelet transform contain the unimportant image features, so quantizing or discarding these coefficients can lead to lossy compression that is both efficient and of high quality. The principle of the Haar transform is to compute averages and differences.

Fractal coding method exploits similarities in different parts of the image. Fractal objects like sierpinski triangle and fern {[1], [2], [3]} have very high visual complexity and low storage information content. For generating computer graphic images and compression of such objects, Iterated Function System (IFS) are recently being used. The basic idea is to represent an image as the fixed points of IFSs and input image can virtually be represented by a series of IFS codes. In short, for fractal coding an image is represented by fractals rather than pixels. Each fractal is defined by a unique IFS consists of a group of affine transformations. Therefore the key point for fractal coding is to find fractals which can best approximate the original image and then to represent them as a set of affine transformations [4]. The traditional FIC is very slow operation; so many methods are used in the paper to speed it up. In this paper the centralized moment is used to sort the blocks of image to find the best matched block. These descriptors also used to predict the symmetry case instead of trying the eight symmetry cases and speed up the compression.

2. YC_bC_r COLOR SPACE

The YC_bC_r color model represents the human perception of color more closely than the standard RGB model, and it is well known that the RGB components of color images are highly correlated and

if the wavelet transforms of each color component is obtained, the transformed components will also be highly correlated.

The YC_bC_r model defines a color space in terms of one luminance (brightness) and two chrominance (color) components. It is one of the most extensively used color spaces and has been considered for many applications. In the YC_bC_r color space, the Y component gives luminance and the C_b and C_r components give the chromaticity values of the color image. To get the YC_bC_r components, the conversion of the RGB components to YC_bC_r components must be known. The RGB to YC_bC_r conversion matrix is given below[5].

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5866 & 0.1145 \\ -0.1688 & -0.3312 & 0.5 \\ 0.5 & -0.4184 & -0.0816 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots(1)$$

To get the RGB values from the YC_bC_r components, the following conversion matrix can be used [5].

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -0.001 & 1.402 \\ 1 & -0.3441 & -0.214 \\ 1 & 1.7718 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} \dots(2)$$

3. Haar Transform

An image is a two-dimensional array of pixel values. The Haar wavelet transform computes the wavelet transform of the image by alternating between rows and columns.

In Haar transform, the image is partitioned into regions such that one region contains large numbers (averages in the case of the Haar transform) and the other regions contain small numbers (differences). However, these regions, which are called subbands, are more than just sets of large and small numbers.

They reflect different geometrical artifacts of the image (Figure 1). The right half of this figure (the differences) is mostly zeros, reflecting the uniform nature of the image. The upper-right subband features traces of the vertical line, whereas the lower-left subband shows traces of the horizontal line. These subbands are denoted by HL and LH, respectively, although there is inconsistency in the use of this notation by various authors). The lower-right

subband, denoted by HH, reflects diagonal image artifacts. Most interesting is the upper-left subband, denoted by LL, which consists entirely of averages. This subband is a one quarter version of the entire image, containing traces of both the vertical and the horizontal lines. It is clear that the lower levels can be quantized coarsely without much loss of important image information, while the higher levels should be quantized finely[6].

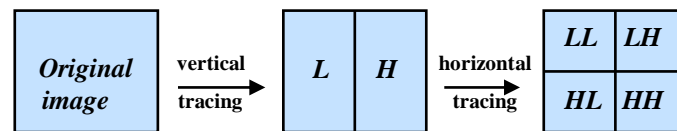


Figure 1: The Pyramid Image Wavelet Transform

4. OVERVIEW OF FRACTAL IMAGE CODING

Fractal image coding can be described as follows: The image to be encoded is partitioned into non-overlapping range blocks Y . The task of the fractal coder is to find a larger block of the same image (a domain block) X for every range block such that a transformation of the domain block is a good approximation of the range block (figure 2). The transformation consists of a geometrical transformation γ and a luminance transformation λ . The geometrical transformation performs a low pass filtering and sub-sampling followed by a position shift. The luminance transformation scales the intensities and changes the mean of the downsampled domain block X . The collage is the approximation that is obtained if all fractal transforms are applied to the original image. Fractal coding consists in finding a good collage that is very similar to the original image. Under the condition that these transformations are contractive, this set of transformations can iteratively be applied to any initial image which then will converge to the decoded image (the fractal attractor). Fractal encoding of images is lossy. Compression can be achieved if the set of transformations can be described more efficiently than the original pixel data. The error between the original image and the fractal collage will always be exceeded by the error of the decoded fractal attractor (Figure 2) [7].

The fundamental principle of fractal image compression consists of the representation of an image by an iterated function system (IFS) of which the fixed point is close to that image. This fixed point is so called 'fractal'. Each IFS is then coded as a contractive transformation with coefficients. Banach's fixed point theorem guarantees that, within a complete metric space, the fixed point of such a transformation may be recovered by iterated implementation thereof to an arbitrary initial element of that space[8].

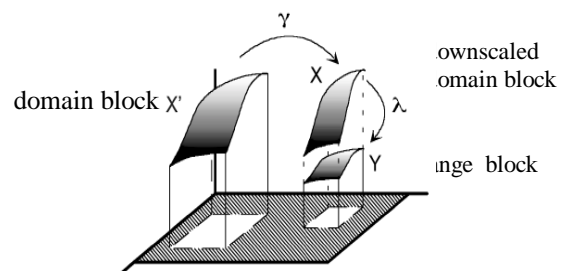


Figure 2: A fractal approximation of a range block Y through a transformed domain block X' [7]

The encoding process is to find an IFS whose fixed point is similar to the given image, is based on the collage theorem, which provides a bound on the distance between the image to be encoded and the fixed point of an IFS. A suitable transformation may therefore be constructed which guaranteeing that the fixed point of that transformation is close to the original image. In the original approach, devised by Barnsley [1], this transformation was composed of the union of a number of affine mappings on the entire image [8]. Among the developed image compression methods is the fractal image coding (FIC) method that based on the theory of partition iterated function systems (PIFS). The scheme of PIFS implies partitioning the image into non-overlapped blocks; these blocks are assembled in range pool. While, the extracted blocks from the down-sampled copy of the image are put in the domain pool. The first step in PIFS compression is achieved if each range block is encoded by providing a reference to one of the domain blocks. The IFS encoding of each range block implies choosing the most resemble domain block listed in the domain pool, and approximating the range by linearly mapping (affine transform) the selected domain block. The main difficulty with PIFS coding method is it takes long time to compress single image[9]. Figure (3) shows the steps of the traditional PIFS encoding system.

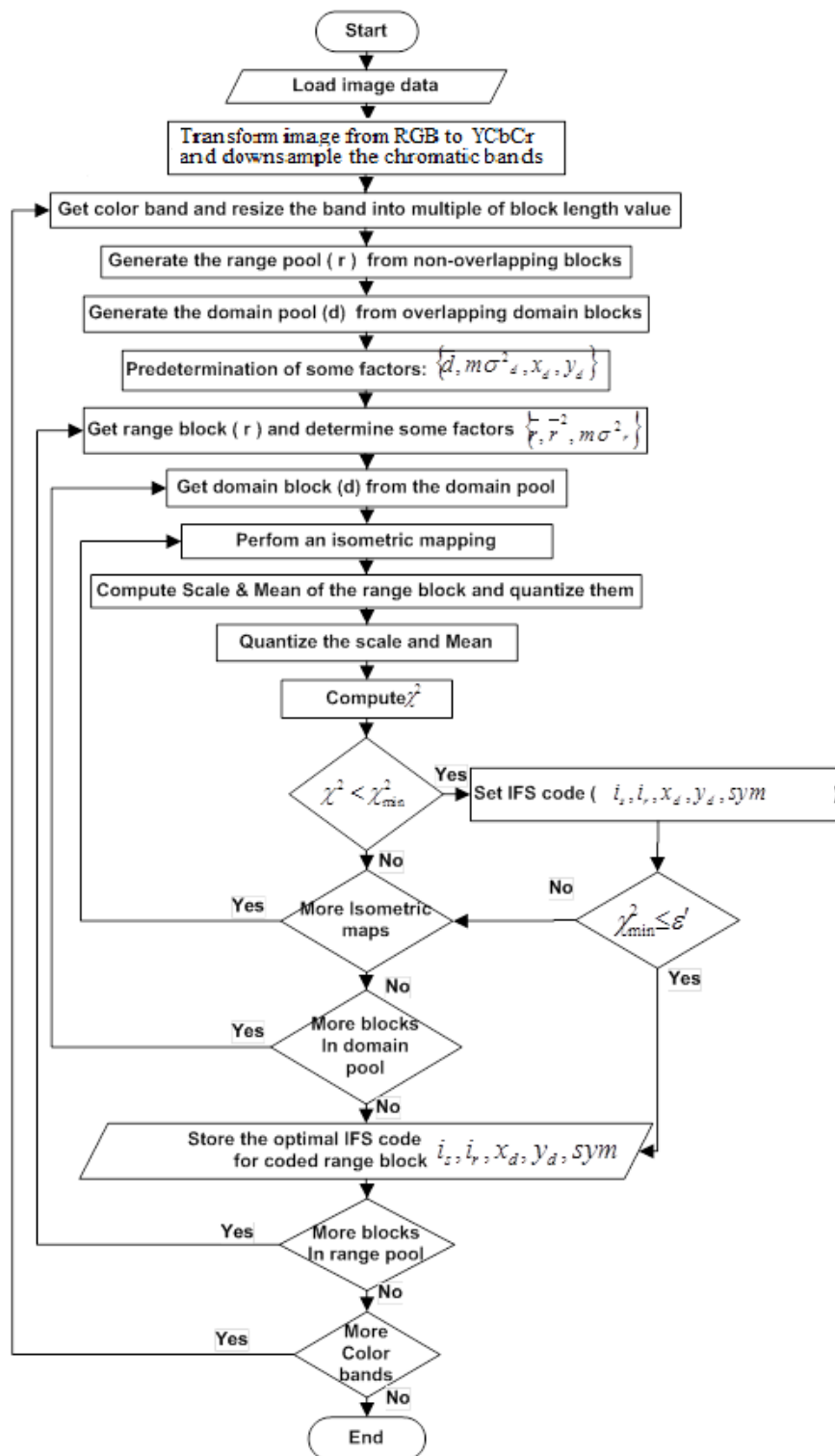


Figure 3: Traditional PIFS system[9]

PIFS image encoder consists of a set of transforms applied on the regions of the image (i.e., range blocks). The transforms are, firstly, used to generate the overlapped domain regions. Secondly, a set of spatial contractive affine transforms are used to approximate the image range blocks by linearly mapping the most similar domain block. For a range block with pixel values $(r_0, r_1, \dots, r_{m-1})$, and the domain

block $(d_0, d_1, \dots, d_{m-1})$, the contractive affine approximation is [10]:

$$r_i = s d_i + o \dots (3)$$

Where s (scale) and o (offset) are the affine transform coefficients, r 's are the approximate (constructed) range values.

The affine transform is changed to become[9]:

$$r'_i = s(d_i - \bar{d}) + \bar{r} \dots\dots\dots(4)$$

Where,

$$\bar{r} = \frac{1}{m} \sum_{i=0}^{m-1} r_i, \quad \bar{d} = \frac{1}{m} \sum_{i=0}^{m-1} d_i \dots\dots(5)$$

Objects are represented as a collection of pixels in an image. Thus, for the purpose of recognition the properties of groups of pixels are needed to be determined. The description is often just a set of numbers (i.e., the object's descriptors). Moments describe the shape's layout (i.e., the arrangement of its pixels), a bit like area. Compactness and irregularity order descriptions.

Moments describe the shape's layout (i.e., the arrangement of its pixels), a bit like area. The calculation of moment invariants for any shape requires knowledge about both the shape boundary and its interior region. The moments used to construct the moment invariants are defined to be continuous but for practical implementation they are computed in the discrete form[11].

For an image block $f(x,y)$ the moment of order (p, q) about the block's center point (x_c, y_c) is defined as [10]:

$$M(p, q) = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y) \dots\dots(6)$$

Apply this definition to determine moments of the domain and range blocks to get [10],

$$M_d(1,0) = \sum_0^{m-1} (x_i - k_c)(d_i - \bar{d}) \dots\dots\dots(7)$$

$$M_d(0,1) = \sum_0^{m-1} (y_i - k_c)(d_i - \bar{d}) \dots\dots\dots(8)$$

$$M_r(1,0) = \sum_0^{m-1} (x_i - k_c)(d_i - \bar{d}) \dots\dots\dots(9)$$

$$M_r(0,1) = \sum_0^{m-1} (y_i - k_c)(d_i - \bar{d}) \dots\dots\dots(10)$$

Where,

$$k_c = \frac{k-1}{2} \dots\dots(11)$$

k is the block width of the image

Now, let us consider the following Moments ratio factor (R)[11]:

$$R = \frac{M^2(0,1) - M^2(1,0)}{M^2(0,1) + M^2(1,0)} \dots\dots(12)$$

The values of the factors (R) are rotation and reflection invariant. Combining equation (4) with equations (7-10), and substitute result in equation (12) we get[10]:

$$R_d = R_r \dots\dots(13)$$

Where:

R_d : is the domain moment-based descriptor value,

R_r : is the reconstructed range moment-based descriptor value.

This above implies that "if any two blocks (from range and domain) satisfy the contractive affine transform, then their moments-based descriptor values should have similar values ($R_n^d = R_n^r$) whatever their isometric state. This does not mean that any two blocks have similar R factors are necessarily similar to each other".

The descriptor is used to sort the domain blocks to reduce the PIFS mapping to be done only with the blocks that have the same or near centralized moment, sorting domain blocks will effectively speed up the matching operation. These moment descriptor is also used to predict the isometric index of the block using the Boolean criterion shown in tables (1 and 2) instead of trying the 8 isometric operations to find the best possible match with the range block [11].

The first widely used signal coding technique was Pulse Code Modulation (PCM), consisting of the independent digitization and coding of signal samples. Limited compression is achievable, however, since inter-sample dependence is completely ignored. The dependence between pixels may be taken into account by coding the prediction error for each pixel, using a prediction based on the values of previously encountered pixels in each scan line. Linear prediction provides a practical alternative only requiring knowledge of the autocorrelation function of the scan lines. The coding of the resulting linear prediction errors is known as Differential PCM (DPCM). DPCM stores only the differences between the information units that are going to be coded except the first information unit [12]. In this paper, DPCM encoding is applied to the fractal coefficients (scale and average) values. At decoding stage, the approximations could be performed several times, starting with any random image, till reaching the fixed point (i.e., attractor state) image. Smooth regions and edges are very self-similar and can be coded efficiently by fractal coders. Irregular textures or noisy regions cannot be approximated well, as they do not possess similarities across scales.

Table 1: The isometric states according of Boolean criterion[11]

State	$Abs(M_x) \geq Abs(M_y)$	$M_x \geq 0$	$M_y \geq 0$
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

Table 2: The introduced predictor output to predict the required symmetry operation [11]

Rot.(Rotation) Ref.(Reflection)	Identity	Rot. (90)	Rot. (180)	Rot. (270)	Ref. at X-axis	Ref. & Rot. (90)	Ref.& Rot. (180)	Ref.& Rot. (270)
Identity	0	6	4	2	5	1	3	7
Rot.(90)	6	0	2	4	1	5	7	3
Rot.(180)	4	2	0	6	3	7	5	1
Rot.(270)	2	4	6	0	7	3	1	5
Ref. at X-axis	5	1	3	7	0	6	4	2
Ref. & Rot. (90)	1	5	7	3	6	0	2	4
Ref. & Rot. (180)	3	7	5	1	4	2	0	6
Ref. & Rot. (270)	7	3	1	5	2	4	6	0

5. PERFORMANCE EVALUATION

To evaluate the performance of the proposed system, the Peak Signal to Noise Ratio (PSNR) based on the Mean Square Error (MSE) is used as a quality measure and its value can be determined using the following equation[16]:

$$PSNR = 10 \log_{10} \left[\frac{255^2}{\frac{MSE(R) + MSE(G) + MSE(B)}{3}} \right] \dots(14)$$

Where

$$MSE = \frac{\sum [f'(m,n) - f(m,n)]^2}{M * N} \dots(15)$$

Here M*N is the total number of pixels in the image, f'(x, y) is the decompressed image and f(x, y) is the original image.

The compression ratio can be measured as the ratio of the number of bits required to represent the image before compression to the number of bits required to represent the same image after compression [17].

6. Methodology

The main objective of the paper is to produce an image compression for color images and suggested three methods that combines wavelet with fractal transform trying to find the best method to compress the color images. To achieve this objective, The original color image is converted from the RGB color space into the YCbCr color space using the appropriate conversion matrix, that is, using equations (1 and 2) each of these components (Y, Cb and Cr) are treated individually in different way in each algorithm. These component are divided into four regions by applying Haar wavelet transform (LL, LH,HL and HH), and then apply the fast fractal image compression FFIC, quantization and DPCM to one (or more) of these regions. Most of the information in the image is found in the LL region which corresponds to the low frequency components of the image and it contains large numbers, because of that it is treated carefully when applying FFIC to keep quality of the reconstructed image, first it is quantized by dividing its value by 8 to have gain in compression ratio, because it gives the same quality when dividing by any number less than 8. The fractal

compression is applied in three different methods as following (Figures 4,5,and 6):

• **First Method:**

Input: RGB Image.

Output: Compressed image file.

Step 1 : The original image is converted from RGB to YCbCr color space.

Step 2: Apply Haar Wavelet transform to Y component.

Step 3: Compress the Y component: The region LL is quantized and saved while the regions (LH,HL and HH) are compressed using FFIC.

Step 4: Apply Haar Wavelet transform to Cb component.

Step 5: Compress the Cb component: The region LL is quantized and saved without any change, while the regions (LH,HL and HH) are ignored and not saved to get more compression ratio, and this will not highly affect the quality of the image because this component already have information about 10% of whole the image information.

Step 6: Apply Haar Wavelet transform to Cr component.

Step 7: Compress the Cr component: It is compressed as same as the component Cb also because of having 10% of the image information.

End.

• **Second Method:**

Input: RGB Image.

Output: Compressed image file.

Step 1: The original image is from RGB to YCbCr color space.

Step 2: Apply Haar Wavelet transform to Y component.

Step 3: Compress the Y component: The region LL is quantized and saved while the regions (LH,HL and HH) are compressed using FFIC.

Step 4: Apply Haar Wavelet transform to Cb component.

Step 5: Compress the Cb component: The LL region is compressed using FFIC transform, while the other three regions are ignored without saving.

Step 6: Apply Haar Wavelet transform to Cr component.

Step 7: Compress the C_r component: The LL region is compressed using FFIC transform, while the other three regions are ignored without saving.
End.

• **Third Method:**

Input: Image transformed with Haar

Output: Compressed image file

Step 1: The original image is converted from RGB to $Y C_b C_r$ color space.

Step 2: Apply Haar Wavelet transform to Y component.

Step 3: Compress the Y component: All regions are compressed using the FFIC.

Step 4: Apply Haar Wavelet transform to C_b component.

Step 5: Compress of C_b component: The LL region is quantized and saved, ignoring the other three regions (LH, HL and HH) without saving.

Step 6: Apply Haar Wavelet transform to C_r component.

Step 7: Compress the C_r component: The LL region is quantized and saved, ignoring the other three regions (LH, HL and HH) without saving.
End.

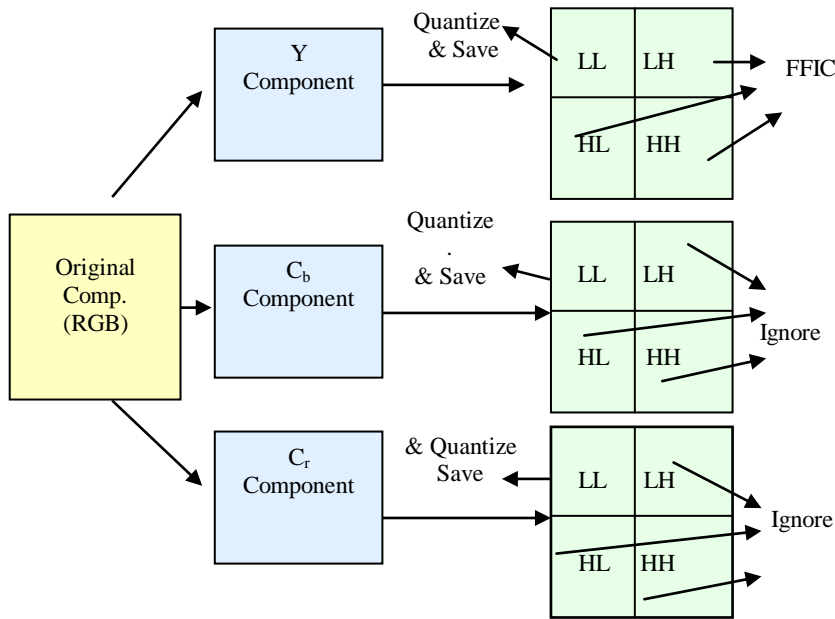


Figure 4 : The first method of the proposed compression structure

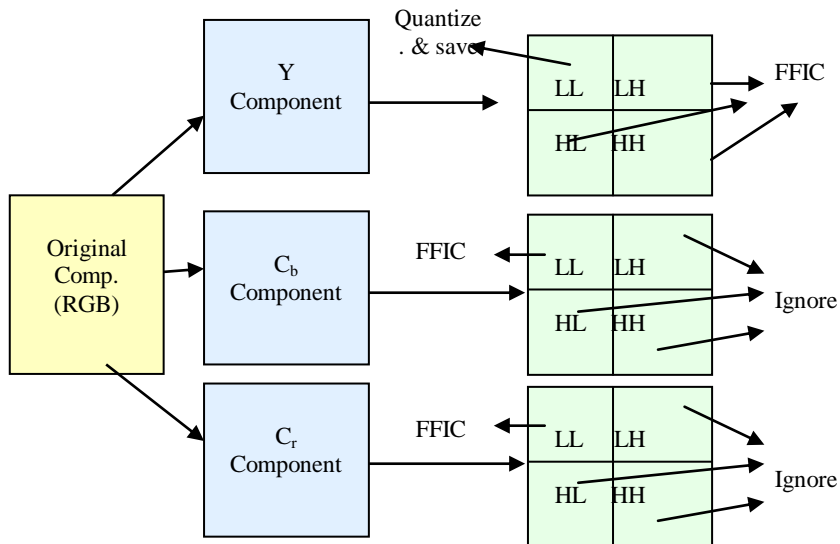


Figure 5: The second method of the proposed compression structure

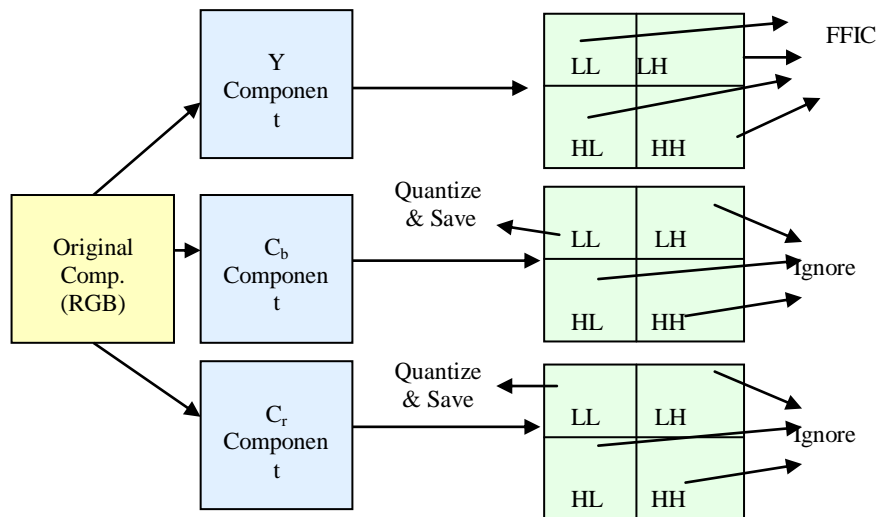


Figure 6: The third method of the proposed compression structure

7. Experimental Results

The performance of the proposed system was rigorously evaluated using quality metrics like Compression Ratio (CR), compression time, decompression time and Peak Signal to Noise Ratio (PSNR). The three proposed methods were applied to the “Lena’s” colored image of size (256x256) (Figure

9), the parameters PSNR, Cr and time are recorded in tables (4, 5 and 6), and compared as shown in figures (7 and 8). The preferred block length in the FFIC was also tested to find the best block length have to be used in all compressed regions (LL, LH, HL and HH) in components (Y, C_b and C_r).

Table 3: Compression parameters before and after applying the proposed methods

	only haar	only Quantization	First method	2'nd method	3'rd method
Cr	0.66	1.7	4.5	10.3	4.8
PSNR	49.4	38.4	28.9	28.7	26.5
Time	1.2	1.2	2.06	2.2	2.4

Table 4: PSNR resulted from applying FFIC with different block length

Component	Tested Region	Used block length				
		64	32	16	8	4
Y	LH	28.7	28.88	28.16	28.08	27.94
	HL	28.7	28.15	28.09	27.95	27.69
	LL	28.7	28.19	28.18	28.16	28.14
C _b	HH	24.91	25.35	25.08	27.75	28.7
C _r	HH	25.27	25.87	27.01	27.73	28.7

Table 5: CR results from applying FFIC with different block length

Component	Tested Region	Used block length				
		64	32	16	8	4
Y	LH	10.3	10.22	10.17	9.84	8.74
	HL	10.3	10.22	10.17	9.84	8.69
	LL	10.3	10.25	10.17	9.86	8.86
C _b	HH	11.72	11.7	11.61	11.28	10.3
C _r	HH	11.72	11.71	11.62	11.31	10.3

Table 6: Time resulted from applying FFIC with different block length

Component	Tested Region	Used block length				
		64	32	16	8	4
Y	LH	2.48	5.78	9.3	10.5	9.2
	HL	2.48	5.64	9.22	11.84	11.86
	LL	2.48	5.67	8.23	8.23	5.80
C _b	HH	2.28	3.02	3.92	3.48	2.48
C _r	HH	2.2	2.5	2.95	2.97	2.48

The tested parameters proved the following results:
 1. The second method is the best method to have good quality with best compression ratio and less time (table 3 and figures 7, 8 and 9).

2. The threshold in the Haar wavelet does not give sensible difference in quality when it changed because of using the FFIC after Haar transform instead or run length encoding.

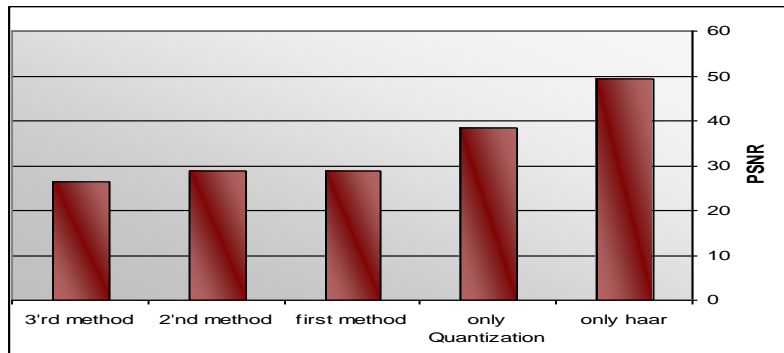


Figure 7: The effect of applying the proposed methods on PSNR

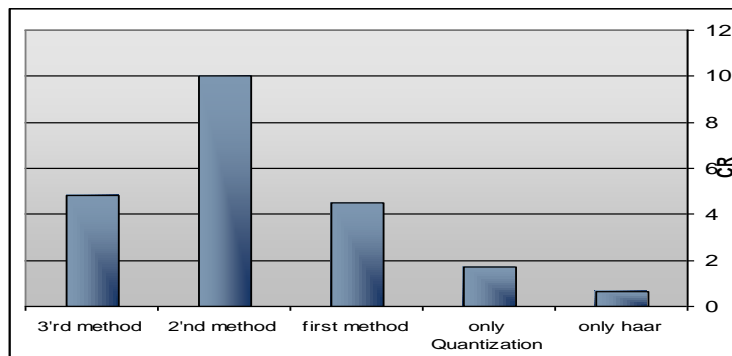
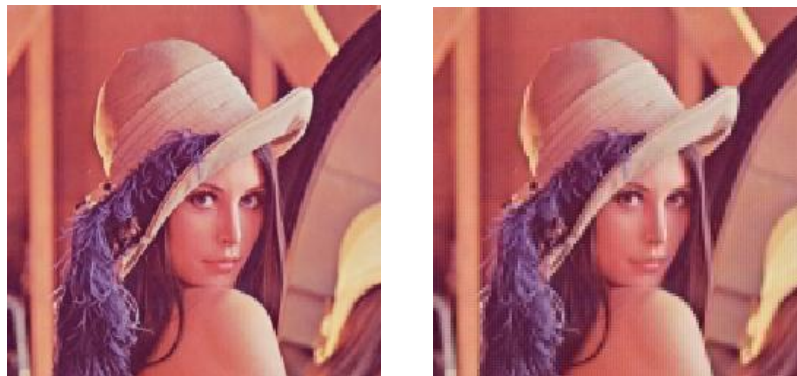


Figure 8: The effect of applying the proposed methods on CR



Lena(256x256)

Reconstructed Lena

Figure 9: Lena’s Image after applying the FFIC –using second method, And it is clear that the reconstructed image have high quality

3. In the Y component regions (HL, LH, and HH), the best block length is 64 with jump step=1, it gives best quality with good CR, because the small blocks will lead to lose some of the important information with best CR and fast compression (table 4 and 5). In the region LL of the components (C_b and C_r), the best block length is 4 with jump step =2 because it speeds up the compression with best quality because of containing little information with small values, so using big blocks will destroy these information when retrieving the fractal coefficients during the decoding operation and this will reduce the quality.

4. The time in Y component regions becomes slow when using small block length, while the (C_b and C_r) components were not highly affected when changing the block length (table 6).

A survey on some of papers that are used both wavelet and fractal transforms and hybrid fractal image compression papers is made (table 7), and the results are compared with the proposed method. It is clear that most of papers have applied their developed system on grey scale images and some of them didn't mentioned the time.

Table 7: The comparison results of the proposed scheme and some previous systems (all applied on Lena image)

Res	Year	Image type	PSNR	CR	Rate	Time
[13]	1995	Gray 512x512	26.81	---	0.08	---
[14]	2006	Gray 512x512	33.90	10	0.84	8.34
[15]	2010	Gray 256x256	28.26	21	0.3	42 sec
[16]	2011	Color 256x256	31.57	11.15	--	---
[17]	2012	Gray 256X256	27.41	---	0.11	8.34 sec
Proposed	2012	Color 256x256	28.7	10.3	---	2.2 sec

8. Conclusions

In the three proposed methods, different results are obtained. The tested results led to the following conclusions:

1. The second method is the best in CR and PSNR. This method leaves the LL in Y component without FFIC compression (because it contains sensitive information) and compresses the other three regions (LH, HL, and HH). In the components (C_b and C_r), the LL is compressed using FFIC transform which rises the CR from 4.5 to 10 without effecting to the quality and time, the other three regions (LH, HL, and HH) are ignored without saving because the components (C_b and C_r) contain little information than Y component especially in the regions (LH, HL, and HH). It can be concluded that in detailed regions of the Haar transformed image, the FFIC support good compression ratio, and reduce the quality, and because of that compressing the LL of the Y component will degrade the quality, so it preferred to save it without any compression, after applying

References

- [1] M. Barnsley, S. Demko, "Iterated Function Systems and the Global Construction of Fractals", Royal Soc., vol. A399, London, 1985.
- [2] M. F. Barnsley, "Fractals Everywhere", Academic Press, 2nd Edition, 1993.
- [3] Yuval Fisher, "Fractal Image Compression", Siggraph'92 Course Notes, 1992.
- [4] Nileshsingh V. Thakur O. G. Kakde,, "Color Image Compression with Modified Fractal Coding on Spiral Architecture", Journal of Multimedia, Vol. 2, No. 4, Aug., 2007.
- [5] H. B. Kekre, Sudeep D. Thepade, and Adib Parkar, "Storage of Colour Information in a

Quantization. In LL of the other two components (C_b and C_r) which contain little details, compressing them with FFIC will rise the CR without noticeable affect to the quality.

2. As known from other papers, the Haar wavelet is a fast transform while the FFIC compression time is too slow, but when applying FFIC on small images or regions will speed up the fractal compression. In this paper the FFIC is applied on the four Haar wavelet transformed image regions with size=1/4 of the image, so the time little affected in the three methods.
3. The block length preferred to be big in the Y component regions (except the detailed region LL which left without FFIC) with jump step=1, and small in the (C_b and C_r) component regions with jump step=2 to make the fractal compression fast.
4. Comparing to other papers (table 7), the proposed paper is the best in encoding time, good quality and accepted compression ratio which can be increased by changing the FFIC parameters like the block length and the jump step.

Grayscale Image using Haar Wavelets and Various Colour Spaces", International Journal of Computer Applications, Vol. 6, No.7, September 2010.

- [6] David Salomon, Giovanni Motta and David Bryant, "Data Compression", Fourth Edition, ISBN-13: 978-1-84628-602-5, Springer-Verlag London Limited 2007.
- [7] K. U. Barthel, H. L. Cycon, D. Marpe, "Image denoising using fractal and wavelet-based methods", Spie, Vol. 5266, P.39-47, 2003.
- [8] Huaqing W., Meiqing W., Tom H., Xiangjian H., and Qiang W., "Fractal Image Compression on a

Pseudo Spiral Architecture", Australian Computer Science Conference, pp 201-208, 2005.

[9] Nevert A., "**Speeding Up Fractal Image Compression Using DCT Descriptors**", Master's thesis, University of Sulaimani, Iraq, 2009.

[10] Loay E. George, "**Fast IFS Coding for Zero-Mean Image Blocks Using Moments Indexing Method**", Iraqi Journal of Science, 2005.

[11] Loay E. George, Suad K. Ahmad, "**Hiding Image in Image Using Iterated Function System (IFS)**", European Conference of Computer Science (ECCS '10), Image, Sound and Signal Processing, Naun/IEEEAM Conferences, 68-74, ISBN: 978-960-474-250-9, 2010.

[12] Wohlberg B., "**Fractal Image Compression and the Self-Affinity Assumption: A Stochastic Signal Modeling Perspective**", Ph.D thesis, University of Cape Town, 2003.

[13] H. Krupnik, D. Malah and E. Karnid, "**Fractal Representation of Images Via The Discrete**

Wavelet Transform", Electrical and Electronics Engineers in Israel, 2.2.2/1 - 2.2.2/5, 1995.

[14] Yuzo Iano, Fernando Silvestre da Silva, and Ana Lúcia Mendes Cruz, "**A Fast and Efficient Hybrid Fractal-Wavelet Image Coder**", IEEE Transaction on Image Processing, vol. 15, No. 1, 2006.

[15] B. Sankaragomathi, L. Ganesan and S. Arumugam, "**Use of Wavelets in Fractal Compression Algorithm for Enhanced Performance**", International Journal of Information Systems for Logistics and Management(IJISLM), Vol. 6, No. 1, 73-80, 2010.

[16] Chandan R. and Sukadev M., "**A Hybrid Image Compression Scheme Using DCT and Fractal Image Compression**", IAJIT first, 2011.

[17] Yih-Lon L. and Wen-Lin C., "**Fast Search Strategies for Fractal Image Compression**", Journal of Information Science and Engineering 28, 17-30, 2012.

تقييم كبس الصور المعالجة بالتحويل الموجي باستخدام الكبس بالكسور المسرعة

نفارت الياس يوسف

المعهد التقني - كركوك ، العراق

(تاريخ الاستلام: 2012 / 2 / 27 ---- تاريخ القبول: 2012 / 9 / 16)

الملخص:

يقدم البحث مجموعة طرق لتطبيق كبس الصور بالكسور المسرعة (FFIC) fast fractal image compression على الصور المعالجة بالتحويل الموجي (Haar Wavelet). اولاً تحول الصورة المستلمة من فضاء الالوان الاحمر، الاخضر والازرق (RGB) الى فضاء الالوان Y, C_b, C_r ، ثم يطبق التحويل الموجي على كل من المكونات الثلاثة Y, C_b, C_r بالتتابع. هذا يكون اربعة اجزاء او حزم جزئية مفترقة وهي: LL, LH, HL, HH . تم اقتراح ثلاث طرق لبيان تأثير كبس الصور بطريقة الكسور المسرعة على هذه المكونات الاربعة. في كل من هذه الطرق يتم معالجة الاجزاء بطريقة مختلفة باستخدام كبس الصور بطريقة الكسور المسرعة على بعض الاجزاء وترك البعض بدون تغيير لإيجاد الطريقة الافضل للكبس. ان الكبس باستخدام الكسور قد تم تسريعه باستخدام مؤشرات العزم المركزي (centralized moment) التي تطبق على كل بلوك من المصدر والهدف، ثم فرز بلوكات الهدف وتقليل عددهم واستخدم ايضا في تحديد زاوية الدوران المناسبة بدون تجربة حالات التدوير الثمانية عند البحث عن افضل تطابق بين بلوكات المصدر والهدف. الاجزاء (HL, LH, HH) في المكونات C_b و C_r تهمل ولا تخزن لزيادة كبس الملف وذلك لكونها لا تحوي معلومات مهمة تؤثر على كفاءة الصورة بينما يعالج الجزء LL من هذه المكونات وكذلك جميع اجزاء المكون Y بشكل مختلف في كل من الطرق المقترحة. تطبق عملية التكميم (quantization) لجميع الاجزاء لتقليل حجم الملف المخزون. واخيرا تم فحص العمل على صورة Lena القياسية (256x256) باستخدام المتغيرات لقياس كفاءة الصورة PSNR، نسبة الكبس (ratio) ووقت الكبس.

الكلمات المفتاحية: Haar Wavelet, Image compression, Fractal Image Compression, color spaces.