



ISSN: 0067-2904

Color Image Compression System by using Block Categorization Based on Spatial Details and DCT Followed by Improved Entropy Encoder

Abdallah A. Ibrahim^{1*}, Loay E. George², Enas Kh. Hassan³

Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Received: 12/1/2020

Accepted: 15/3/2020

Abstract

In this paper, a new high-performance lossy compression technique based on DCT is proposed. The image is partitioned into blocks of a size of $N \times N$ (where N is multiple of 2), each block is categorized whether it is high frequency (uncorrelated block) or low frequency (correlated block) according to its spatial details, this done by calculating the energy of block by taking the absolute sum of differential pulse code modulation (DPCM) differences between pixels to determine the level of correlation by using a specified threshold value. The image blocks will be scanned and converted into 1D vectors using horizontal scan order. Then, 1D-DCT is applied for each vector to produce transform coefficients. The transformed coefficients will be quantized with different quantization values according to the energy of the block. Finally, an enhanced entropy encoder technique is applied to store the quantized coefficients. To test the level of compression, the quantitative measures of the peak signal-to-noise ratio (PSNR) and compression ratio (CR) is used to ensure the effectiveness of the suggested system. The PSNR values of the reconstructed images are taken between the intermediate range from 28dB to 40dB, the best attained compression gain on standard Lena image has been increased to be around (96.60 %). Also, the results were compared to those of the standard JPEG system utilized in the "ACDSee Ultimate 2020" software to evaluate the performance of the proposed system.

Keywords: Image Compression, Image Coding, Discrete Cosine Transform, Entropy Coding, Quantization.

ضغط الصور الملونة باستخدام طريقة تصنيف البلوك بالاعتماد على الخصائص المكانية للبلوك وطريقة التحويل المويجي (DCT) متبوعة بطريقة الترميز الانتروبي (Entropy Coding) المحسن

عبدالله عادل ابراهيم*, لؤي ادور جورج, ايناس خزعل حسن

قسم علوم الحاسوب, كلية العلوم, جامعة بغداد, العراق

الخلاصة

في هذا البحث، تم تطبيق نظام جديد لضغط الصور باستخدام طريقة التحويل المويجي (DCT). في هذا النظام يتم تقسيم الصورة الى بلوكات صغيرة بحجم $N \times N$ (بحيث N يجب ان يكون مضاعفات العدد 2)، ثم يتم تصنيف البلوك بالاعتماد على التفاصيل المكانية (Spatial Redundancy) الموجودة فيه وذلك بحساب طاقة البلوك عن طريق حساب مجموع الفرق بين البكسلات الموجودة في هذا البلوك (DPCM) لتحديد اذا كان الترابط بين البكسلات لهذا البلوك عالي او قليل. بعدها يتم تحويل البلوك الى مصفوفة احادية

*Email: Prog.abdallah@gmail.com

بعد ذلك من اجل سرعة العمليات ، وذلك باستخدام القراءة الاقفية المتعامدة للقيم الموجودة في البلوك. بعد هذا يتم تطبيق طريقة التحويل الجيب تمام احادية البعد (ID-DCT) لتحويل البلوك الى صيغة اكثر كفاءة للتعامل معه. ثم يتم تكميم (Quantization) المعاملات الناتجة بعد التحويل باستخدام قيم تكميم مختلفة بالاعتماد على اهمية البلوك. في المرحلة الاخيرة يتم تطبيق طريقة الترميز الانتروبي (Entropy Coding) المحسن لخرن بيانات الصورة بعد الضغط من اجل النقل او الاسترجاع . تم تقييم النظام على عدة صور نموذجية باستخدام طرق التقييم (MSE,PSNR,CR)، قيمة ال PSNR تم اخذها ما بين المتوسط المقبول من (28dB الى 40dB) وقد تم الحصول على نسبة ضغط عالية بما يقارب 96,60% عند تطبيق النظام على صورة لنا، ايضا تم مقارنة النتائج مع نتائج نظام (JPEG) الرسمي المستخدم في تطبيق " ACDSsee Ultimate 2020" من اجل تقييم النظام .

1. Introduction

Data Compression means to compress data to reduce the data storage needed to store it and also reduce transmission time needed to transmit it over the Internet. Data is consisting of information and redundancy. The information must be continuously preserved in its original form to properly understand the meaning of the data while redundancy can be removed when it is not important to reduce data size or reinsert it again to data to return it to its original form when needed. The technique of removing redundancy is called a compression and the technique of reinserting redundancy is called decompression [1].

There are varieties of redundancies that can be eliminated from the color image, these redundancies are supposed to be redundant information and can be removed, these include (Spectral redundancy, Inter pixel redundancy, Psycho-visual redundancy, and Coding redundancy,). One of the main issues of RGB system is the spectral redundancy that usually exists between pixels of the three primary color channels. In order to provide a more efficient representation of color images, the YCbCr (Luminance, Blue Chrominance and Red Chrominance) color system, also known as YUV, has been proposed in ITU-R BT.601 Recommendation [2]. Inter pixel redundancy presents because of the correlation between pixels of an image, each pixel is related to its neighbor's pixels and its value can be predicted from its neighbors, this sort of redundancy is likewise called (spatial redundancy). Psycho-visual redundancy occurs in image because of the Human visual system (HVS), human eye cannot distinguish all image colors because "it is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum", removing less important frequencies (information) in the image may be acceptable to human visual system. Coding redundancy occurs in an image because of the probability of the same intensity occurrences, to reduce that type of redundancy use shorter code words for most occurring intensity values and longer code words to the least occurring ones, this is called (variable length coding) [3], [4].

In this work, high performance color images compression system was implemented, the system comprises of four stages to deal with different type of redundancies that have been mentioned earlier. The first stage used to transform color space from RGB to $YCbCr$, to reduce spectral redundancy. The second stage is the transform coding to deal with inter pixel redundancy. The third stage is the quantization to deal with psycho-visual redundancy and the final stage which is the most important stage in this work is the entropy coding to deal with coding redundancy.

2. Literature Survey

Ashwaq T. Hashim et al. proposed a compression system based on DPCM, DCT, DWT and Quad-tree coding techniques. Firstly, the RGB space is converted into YUV color space to reduce spectral redundancy. Secondly, the image is partitioned into blocks and the DPCM technique is used to separate blocks into correlated and non-correlated blocks by finding the Maximum absolute value. Then DCT Is used to transform correlated blocks and DWT used to transform non-correlated blocks. Then quad-tree coding technique applied to uncorrelated blocks to produce the output sequence. The resulted sequence from quad-tree coding and the resulted coefficients from DCT is coded and stored using a shift coding technique. The performance results of proposed algorithms produce a better quality of the image in terms of PSNR with a higher CR compared to standalone DCT or DWT [5].

Yangming Zhou et al. propose a lossy compression algorithm using a new efficient lossless encoder. Firstly, the image is converted into YCbCr color space, then DCT applied to transform an image and produce the transformed coefficients. An iterative process based on the bisection method

used to control the quality of compressed image, this algorithm applies the bisection method to repeat an iterative process to automatically select the threshold. After setting the goal of the peak signal-to-noise ratio. An adaptive block scanning by using four different aspects of scans (zigzag, horizontal, vertical, and Hilbert) applied to each transform coefficients block by choosing the path which has the longest run of zeros. The next step is to encode the transformed block using a modified lossless encoder, which is work by utilizing the statistical characteristics of the DCT coefficients. The proposed system has a better performance compared to JPEG and CDABS for both subjective and objective measurements [6].

Abdelhamid Messaoudi et al. present a compression algorithm based on DCT. First, the RGB color space is transformed into a YCbCr color space. Next, the image is partitioned into blocks of equal size and DCT applied on each block. Four types of scan orders applied on each block (Zig-Zag, Horizontal, Vertical, and Hilbert), then an index vector contains the runs of zero sequence is formed for each scan. The lowest length of the four index vector is determined as the best scan for a block. The non-zero values and the chosen index vector are coded and stored. The proposed system shows better performance compared to several techniques including JPEG standard [7].

Mohammed M. Siddeq et al. propose a new system for image compression based on DCT. The method works by split the image into blocks and then the DCT applied for each block to produce the transformed coefficients. Then each block is converted into a 1D vector. After that, the zero values from AC coefficients are removed and keep only non-zero values, then apply a high-frequency minimization method to the non-zero values reducing each block by 2/3 and this will produce a minimized array. A Differential operator is applied to the vector of DC-components. Finally, the resulted DC and AC coefficients are stored using arithmetic coding. The attained results show that the proposed compression system is perceptually superior to JPEG with equivalent quality to JPEG2000 [8].

Issam Dagher et al. use a combination of DCT and Haar wavelet transform is done to make use of the advantages of these two transforms. The image is partitioned into an equal-sized block, then DCT is used to transform the upper left corner of the block and the Haar is used to transform the remaining parts of the block. The system results show an improvement in PSNR value with the same CR compared to DCT, and permitting better edge recovery than the Haar transform [9].

3. The Proposed System Scheme

Firstly, the color transformation from RGB space to $YCbCr$ is applied. Then, the input image is split into blocks of size $N \times N$ (where N is multiple of 2), each block is categorized according to its spatial details whether it is high-frequency block (its pixels are uncorrelated) or low-frequency block that contains correlated pixels by using DPCM in three different aspects (horizontally, vertically, and diagonally) then the energy of each block is calculated to determine the correlation level between its nearby pixels by using a specified threshold value. Then, image blocks are scanned and converted into 1D vectors using horizontal scan order. The next step is to apply the 1D-DCT on each vector to produce the transform coefficients. Then adaptive scalar quantization applied for both correlated and uncorrelated blocks. The level of the quantization values for each block is different according to the block feature whether it is high frequency detailed blocks or low frequency correlated blocks, by taking advantages that these high frequencies blocks can be treated separately from low frequency to produce better compression. Finally, the proposed entropy encoder applied to the quantized coefficients to store each coefficient with an optimal number of bits. Figure- 1 illustrates the process.

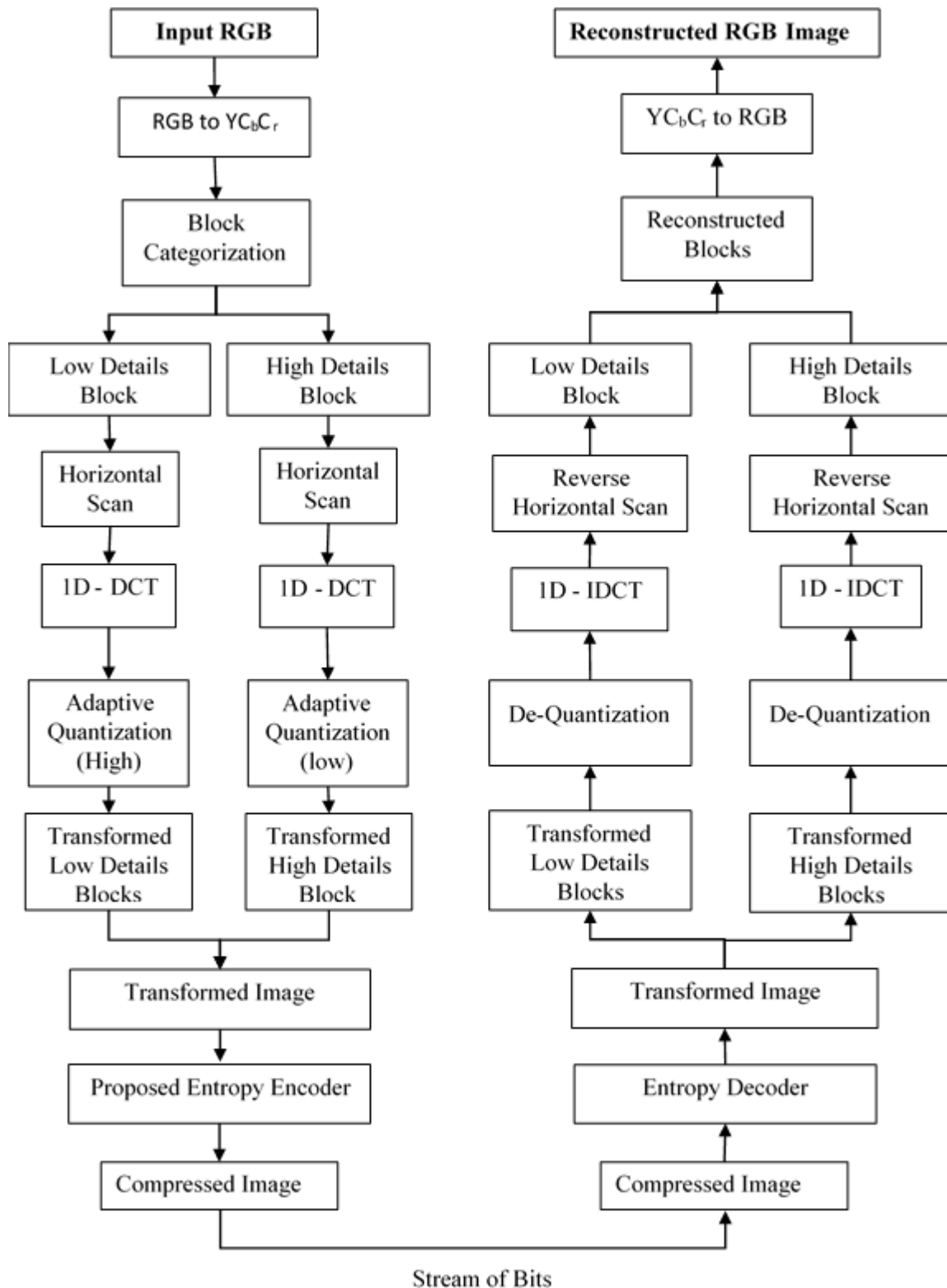


Figure 1- Block diagram of the encoder and decoder units

3.1 RGB to YCbCr Conversion

Firstly, RGB color space is converted to YCbCr space. One of the main issues of the RGB color space is the spectral redundancy that usually exists between pixels of the three primary color channels. To provide a more efficient representation of color images, the YCbCr space (luminance component and chrominance component) Y represents luminance information and C_b, C_r is chrominance

information. The logic behind this process is that most of the image energy lies within the Y component and also human eye is more sensitive towards luminance change than color changes. So is beneficial to work in the YCbCr space and to treat these three components separately.

The transform from RGB color space to YCbCr space is given by the following equations [10]:

$$Y = 16 + \left(\frac{65.738 \text{ Red}}{256}\right) + \left(\frac{129.057 \text{ Green}}{256}\right) + \left(\frac{25.064 \text{ Blue}}{256}\right) \quad (1)$$

$$Cb = 128 + \left(\frac{-37.945 \text{ Red}}{256}\right) - \left(\frac{74.494 \text{ Green}}{256}\right) + \left(\frac{112.439 \text{ Blue}}{256}\right) \quad (2)$$

$$Cr = 128 + \left(\frac{112.439 \text{ Red}}{256}\right) - \left(\frac{94.154 \text{ Green}}{256}\right) - \left(\frac{18.285 \text{ Blue}}{256}\right) \quad (3)$$

The inverse transform from YCbCr color space to RGB space is given by the following equations:

$$\text{Red} = \left(\frac{298.082 Y}{256}\right) + \left(\frac{408.583 Cr}{256}\right) - 222.921 \quad (4)$$

$$\text{Green} = \left(\frac{298.082 Y}{256}\right) - \left(\frac{100.291 Cb}{256}\right) - \left(\frac{208.120 Cr}{256}\right) + 135.576 \quad (5)$$

$$\text{Blue} = \left(\frac{298.082 Y}{256}\right) + \left(\frac{516.412 Cb}{256}\right) - 276.836 \quad (6)$$

3.2 Block Classification

Image blocks are categorized by applying differential pulse code modulation (DPCM) on each block to determine the correlation between its pixels and hence determine the importance of block. The DPCM for each block is calculated in three different manners (Horizontally, Vertically and Diagonally) to calculate the correlation of each pixel with its neighborhood pixels. Then the level of correlation of this block is determined by taking the maximum absolute sum of these differences. If the correlation level of the block is greater than a specified threshold, then this block is treated as high-frequency uncorrelated block, otherwise the block is treated as low frequency correlated block.

The correlation level of each block is calculated using the following equations:

$$\text{Horizontal} = \frac{\sum_{i=0}^{w,h} \text{Abs}(Y(i,j) - Y(i+1,j))}{wxh} \quad (7)$$

$$\text{Vertical} = \frac{\sum_{i=0}^{w,h} \text{Abs}(Y(i,j) - Y(i,j+1))}{wxh} \quad (8)$$

$$\text{Diagonal} = \frac{\sum_{i=0}^{w,h} \text{Abs}(Y(i,j) - Y(i+1,j+1))}{wxh} \quad (9)$$

where:

i, j: is the coordinate of each pixel in the block.

Y: is the pixel value with the coordinates i,j.

w, h: width and height of the block.

3.3 Horizontal Scanning Order

The horizontal scan is used to convert the 2D block into 1D vector. Horizontal scan working by scanning each row in snake form. For first row, the scan is starting from first pixel into last pixel and in the second row the scan is starting from last pixel into first pixel and so on. The horizontal scan is shown in Figure-2.

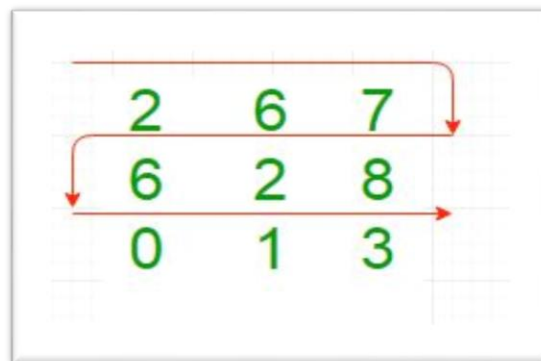


Figure 2- Horizontal scan order on a sample image block

3.4 Discrete Cosine Transform (DCT)

DCT implemented in this system is a 1D-DCT. Because processing the block in one-dimension manner is faster than when processed in two dimensions, and therefore accelerating the compression and decompression process. The resultant DCT transformed coefficients in each block is arranged in a way where the coefficient that holds the most significant information is called DC component and it resides in the upper left corner of the block, the rest of the coefficients holds the less important information in the block which is called AC [11].

1D-DCT can be obtained by applying the equation (10) to transform the images and reverse the transformation by applying the equation (11) to obtain the reconstructed image [12]:

$$F(u) = C(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \quad (10)$$

Where, N is the length of the input block, $u = 0, 1, \dots, N-1$

$$C(u) = \sqrt{\frac{1}{N}} \quad \text{when } u=0 \quad C(u) = \sqrt{\frac{2}{N}} \quad \text{when } u \neq 0$$

$$f(x) = C(u) \sum_{u=0}^{N-1} F(u) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \quad (11)$$

$$C(u) = \sqrt{\frac{1}{N}} \quad \text{when } u=0 \quad C(u) = \sqrt{\frac{2}{N}} \quad \text{when } u \neq 0$$

Where $f(x)$ the input matrix.

3.5 Adaptive Scalar Quantization

Quantization is a simple process, each coefficient is divided by a quantization step value, and then the result will be rounded to the nearest integer value, the de-quantize process is carried out by multiplying with quantization step. In non-uniform scalar quantization, the quantization matrix of size $N \times N$ is calculated, then each transformed block of size $N \times N$ from the DCT step is quantized non-uniformly by dividing the coefficients of each block by its corresponding value in the quantization matrix. Quantization process is applied by the following equation [13]:

$$DCT_Q(x, y) = \text{round} \left(\frac{DCT_c(x, y)}{Q_s} \right) \quad (10)$$

where the quantization step (Q_s) is calculated by the following equation:

$$Q_s(u, v) = \begin{cases} Q_0 & \text{for DC coefficient} \\ Q_1 * (1 + (\alpha * (u + v))) & \text{for AC coefficients} \end{cases} \quad (11)$$

Where:

u, v : represents the coordinates of each block

Q_0 is calculated as follows:

$$Q_0 = \frac{N \times N}{2}$$

Where N is the block size.

4. Entropy Coding

An entropy coding applied to store the quantized coefficients as the compressed image for less storage requirement. The entropy encoder work by using variable length coding (VLC) technique, VLC technique calculates the optimal number of bits needed for each coefficient according to their probability. Each quantized block will be passed into the encoder to produce a stream of bits.

The algorithm steps of the proposed entropy encoder implemented as the following:

1- Split the image into a block of size ($M \times M$), M is calculated as the block size (N) used by the DCT stage divided by 2.

2- Check if the block values are all zeroes or contain zero and non-Zero values, a tag with one bit (0 or 1) is stored for each block to use it later in the decompression stage to determine that the corresponding block elements are all zeroes.

- 3- If the block values are all zero then the block is discarded, whereas if the block contains non-Zero values, create a vector that will contain block values sequentially. For DCT block take the DC value until the last non-zero value found and discard the rest of the block.
- 4- Map the negative values into positive using mapping operation.
- 5- Move the non-zero values in the resulted vector and put it in a new vector and replace its location in the original vector with one.
- 6- The run-length encoder applied for the original vector (contains only zeroes and ones) to reduce the long-run sequence.
- 7- Find the mean value of each resulted two vectors (Non-Zero vector and Run vector).
- 8- For each vector, move the values that are equal or greater than its mean value and put it in a new vector and replace its original location with zero.
- 9- Subtract the values in the resulted vectors from the mean value.
- 10- Adaptive shift coding applied for each vector to specify the optimal number of bits needed to store each value.

4.1 Mapping

The mapping process is required to convert the negative values into positive numbers to get rid of the negative sign and also avoid the coding complexity when storing these numbers. This is simply done by mapping all negative values to positive odd numbers while the positive values are mapped to be even numbers. The mapped numbers are produced by applying the following equation [14]:

$$P_i = \begin{cases} 2 P_i & \text{if } P_i \geq 0 \\ (|P_i| * 2) - 1 & \text{if } P_i < 0 \end{cases} \quad (14)$$

Where P_i is the coefficient value in the incoming sequence.

The inverse mapping process is implemented using the following equation:

$$P'_i = \begin{cases} P_i / 2 & \text{if } P_i \text{ is even number} \\ (-P_i + 1) / 2 & \text{if } P_i \text{ is odd number} \end{cases} \quad (15)$$

4.2 Zeroes and Non-Zeroes Decomposition

In this stage, each block is separated into two sequence vectors. Firstly, take the Non-Zero values in the block and put it in new vector in sequence and replace its original location in the block with ones, Secondly, take the zeros and ones in the block and put it in new vector. The following example illustrate the process.

Let us consider the following image block:

25	15	9	1	0
12	11	2	0	0
10	1	0	0	0
1	0	0	0	0
0	0	0	0	0

Figure 3- Sample Image Block

By taking the non-zero values and replacing its location with ones this will results in:

1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0
0	0	0	0	0

Figure 4- Sample Image Block (Zeroes and Ones)

Separating block into two sequence vectors this will results in:

Zeroes and Ones Vector:

1,1,1,1,0,1,1,1,0,0,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0

Non-Zero Vector:

25,15,9,1,12,11,2,10,1,1

4.3 Run Length Encoding (RLE)

RLE is applied on a 1D-vectors that contains a lot of consecutive zeroes. We can reduce the size of the vector by replacing a sequence of the same values with only one value along with its count. Let us consider the following sequence with zeroes and non-zeroes values:

57, 45, 0, 0, 0, 0, 23, 0, -30, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, ... , 0

The run-length encoding is applied in the following steps:

1- By replacing the non-zero values with ones and storing them in a different vector, we obtain:

1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, ... , 0

57, 45, 23, -30, -16, 1

2- A run-length encoding will be applied to the sequence of zeroes and ones, as follows:

(1,2), (0,4), (1,1), (0,1), (1,2), (0,2), (0,EOB)

3- We can discard the last zero-sequence because it is already known that the values of the remaining blocks are zeroes:

(1,2), (0,4), (1,1), (0,1), (1,2), (0,2)

4- Only the non-zero values and the length of each sequence will be coded and stored:

57, 45, 23, -30, -16, 1 (non-zero values)

2, 4, 1, 1, 2, 2 (Run length of each sequence)

4.4 Mean Value Decomposition

In this stage, the resultant vectors from the previous stage (Non-Zero Vector and Run Vector) is further separated into two sub-vectors according to the mean value. The first vector will contain the values less than the mean value and the second vector will contain the values that is greater than or equal to the mean value and then subtract the second vector from the mean value to reduce its scaling range.

4.5 Adaptive Shift Coding

Shift coding is one of the variable length coding (VLC) technique. Shift coding efficiently removes coding redundancy in the sequence by assigning an optimal number of bit needed per value according to their probability. Most frequent occurring coefficients will be assigned a short number of bit than less occurring coefficients. Shift coding working by determining the minimum total bits needed to store all values in the input sequence. The following equations calculate the total number of consumed bits according to the nature of the input data [15]:

$$T_{Bits} = (n_s + 1) \sum_{i=0}^{i=2^{n_s}-1} His(i) + (n_l + 1) \sum_{i=2^{n_s}}^{\max(His)} His(i) \quad (16)$$

$$\text{or } T_{Bits} = n_s \sum_{i=0}^{i=2^{n_s}-1} His(i) + (n_l + n_s) \sum_{i=2^{n_s}}^{\max(His)} His(i) \quad (17)$$

where (n_s) is the number of bits required for short values and (n_l) is the number of bits required for long values, while $His()$ represents the histogram array for the input sequence. Equation (16) is applied to find the minimum T_{Bits} using shift key{1,0} as keys indicator (i.e., the values 1 and 0 are stored as an additional bit for indicating that the value is stored with a short or long number of bits). While equation (17) is employed in the case of using the value (n_s) as a key-value without the need for using extra key indicators. The values of n_s & n_l are chosen such that they lead to minimum T_{Bits} . After determining the optimal values of n_s and n_l , the most frequently appearing data will be stored in a short number of bits (n_s) while the less frequently appearing data will be stored in a long number of bits (n_l) , along with its key indicator.

5. Evaluation Metrics

The Compression Ratio (CR) is defined as the ratio of the size between the original image and the compressed image. It can be measured in bits, bytes, kilobytes, etc.... The higher CR value is the better compression technique used to compress the image [16].

$$CR = \frac{\text{Original Size}}{\text{Compressed}} \quad (18)$$

Mean Square Error (MSE) is one of the "full-reference" evaluation metric and the simplest estimator to measure image quality. MSE is computed by averaging the cumulative squared intensity error differences between the original image and the reconstructed image. MSE value closest to zero is better, the reconstructed image quality is poor when MSE is large, thus MSE must be as low as possible for effective compression. MSE is calculated through the following equation [17]:

$$MSE = \frac{1}{m n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [X(i, j) - Y(i, j)]^2 \quad (19)$$

where X is the original image size and Y is the compressed image size in pixel. The dimension of the images is m x n.

Peak Signal to Noise Ratio (PSNR) is the ratio between maximum signal power which is considered as the original image and the power of distorting noise obtained from MSE. PSNR measure how the human visual reaction to image quality.

The higher the PSNR, the better the quality of the reconstructed image. Typical values for lossy compression of an image are between 28 dB and 40 dB. The PSNR can be given by the following equation [18]:

$$PSNR = 10 \log_{10} \left(\frac{MAX_x^2}{MSE} \right) \quad (20)$$

where MAX_x^2 is the power of maximum intensity value in the original image X.

Bit Per Pixel (BPP) is defined as a number of bits required to store each pixel in an image. Pixels must be coded efficiently to reduce redundancy, hence reduce storage requirements. BPP is calculated using the following equation [19]:

$$BPP = \frac{B}{M \times N} \quad (21)$$

$$\text{or } BPP = \frac{24 \text{bits}}{CR} \quad (22)$$

where B is a number of bits after compression and MxN is the total number of pixels in an image. The CG is defined as the amount of compression gained after the image is compressed. The CG of a digital image is calculated by the following equations:

$$CG = \frac{\text{original size} - \text{compressed size}}{\text{original size}} \times 100\% \quad (23)$$

$$\text{or } CG = 1 - \frac{1}{CR} \times 100\% \quad (24)$$

The Structural similarity index Measure (SSIM) metric aims to measure quality by capturing the similarity of images. Three aspects of similarity: Luminance, contrast and structure is determined and their product is measured.

6. Test Materials

A set of standard true color images have been utilized to test and evaluate the system performance. These images are of type "Bitmap" format in which each pixel is stored in 24-bit true color. The image files "Lena.bmp", "Barbara.bmp", "Peppers.bmp", (smoothed images) and "Baboon.bmp" (sharp edge image). All test images are of size 256x256 for width and height. Figure- 5 presents these images.



A. Color Lena

B. Color Barbara

C. Color Peppers

D. Color Baboon

Figure 5- Test color images

7. Experimental Results

In this section, a set of results table are presented for evaluation. Each table shows the effectiveness of each control parameter used in the proposed system on the resulted compressed image. The test results are evaluated and compared based on fidelity criteria measurements (i.e. MSE, PSNR, SSIM, CR, CG and BPP). The PSNR values in each table ranged from 28 dB (low quality) to 40 dB (very high quality). The real time of encoding and decoding process also presented. The adopted system and all additional programs for the testing purpose were established using visual studio (C sharp programming language).

The effect of the following control parameters has been investigated to test the results of the proposed system:

1. **BS**: is the block size used in DCT
2. **Thr.**: is the threshold value used to determine the importance of each block.
3. **Q0, Q1, and α** : are the quantization control parameters used to calculate the quantization step value.

The selection of each parameter depends on the characteristics of the tested image. The range of each control parameters is shown in the table below.

Table 1- The default range for control parameters of the system

BS	Thr.	Q0	Q1	α
8	[1-25]	32	[1 - 10]	[0.1 - 1]
16	[1-25]	128	[1 - 10]	[0.1 - 1]

Table 2- Test results for color Lena image

BS	Thr.	MSE	PSNR(dB)	SSIM	CR	CG (%)	BPP	Time in (Sec.)	
								ET	DT
16	20	101.760	28.055	0.853	29.432	96.602	0.815	0.564	0.430
8	10	96.144	28.302	0.851	22.609	95.577	1.062	0.207	0.158
16	10	81.606	29.014	0.885	21.933	95.441	1.094	0.468	0.390
8	10	80.547	29.070	0.870	20.052	95.013	1.197	0.300	0.223
8	20	63.814	30.082	0.897	17.111	94.156	1.403	0.235	0.189
8	20	51.104	31.046	0.917	14.234	92.974	1.686	0.198	0.163
8	5	37.733	32.364	0.944	11.303	91.152	2.123	0.207	0.159
8	15	32.120	33.063	0.955	9.690	89.680	2.477	0.204	0.163
8	10	25.666	34.037	0.967	7.990	87.485	3.004	0.209	0.162
8	25	20.231	35.071	0.978	6.272	84.056	3.827	0.274	0.193
8	5	15.883	36.121	0.985	4.609	78.305	5.207	0.239	0.194

Table 3- Test results for color Barbara image

BS	Thr.	MSE	PSNR(dB)	SSIM	CR	CG (%)	BPP	Time in (Sec.)	
								ET	DT
16	25	111.797	27.646	0.859	29.948	96.661	0.801	0.627	0.515
16	20	102.298	28.032	0.869	27.356	96.345	0.877	0.469	0.390
8	5	108.890	27.761	0.853	22.309	95.517	1.076	0.196	0.150
8	5	101.707	28.057	0.860	21.629	95.377	1.110	0.198	0.150
8	5	80.717	29.061	0.885	19.224	94.798	1.248	0.199	0.152
8	25	64.805	30.015	0.901	17.104	94.153	1.403	0.201	0.161
8	25	51.157	31.042	0.921	15.281	93.456	1.571	0.243	0.195

8	20	40.089	32.101	0.938	13.321	92.493	1.802	0.202	0.162
8	15	32.530	33.008	0.948	10.815	90.754	2.219	0.271	0.231
8	25	25.325	34.095	0.963	9.183	89.111	2.613	0.215	0.162
8	25	20.533	35.006	0.971	7.972	87.456	3.010	0.273	0.209
8	25	16.056	36.074	0.980	6.667	85.001	3.600	0.262	0.211

Table 4- Test results for color Peppers image

BS	Thr.	MSE	PSNR(dB)	SSIM	CR	CG (%)	BPP	Time in (Sec.)	
								ET	DT
8	25	135.689	26.805	0.851	16.816	94.053	1.427	0.247	0.189
16	25	141.150	26.634	0.852	16.086	93.784	1.492	0.465	0.392
8	20	129.587	27.005	0.856	16.064	93.775	1.494	0.204	0.163
16	25	124.290	27.186	0.869	13.702	92.702	1.752	0.480	0.384
8	25	102.116	28.040	0.881	13.342	92.505	1.799	0.201	0.155
8	5	79.398	29.133	0.903	10.370	90.356	2.314	0.216	0.167
8	25	64.014	30.068	0.927	8.308	87.963	2.889	0.207	0.160
8	5	49.674	31.169	0.941	6.787	85.266	3.536	0.217	0.166
8	5	38.472	32.279	0.958	5.474	81.733	4.384	0.218	0.174
8	5	29.408	33.446	0.975	4.288	76.679	5.597	0.226	0.175
8	5	24.084	34.314	0.984	3.556	71.879	6.749	0.268	0.209
8	5	17.985	35.582	0.990	2.924	65.806	8.207	0.268	0.244

Table 5- Test results for color Baboon image

BS	Thr.	MSE	PSNR(dB)	SSIM	CR	CG (%)	BPP	Time in (Sec.)	
								ET	DT
16	5	309.909	23.218	0.854	10.972	90.886	2.187	0.726	0.507
8	10	276.590	23.712	0.850	9.586	89.568	2.504	0.205	0.162
16	15	254.783	24.069	0.878	9.477	89.448	2.532	0.589	0.478
8	15	251.498	24.125	0.853	9.208	89.139	2.607	0.247	0.194
8	15	202.882	25.058	0.893	8.049	87.576	2.982	0.212	0.164
16	10	193.000	25.275	0.927	7.492	86.652	3.204	0.492	0.386
8	10	162.418	26.024	0.930	6.649	84.959	3.610	0.253	0.207
8	5	127.801	27.065	0.961	5.189	80.727	4.625	0.257	0.219
8	5	102.496	28.024	0.975	4.187	76.118	5.732	0.293	0.256
8	5	73.284	29.481	0.985	3.362	70.253	7.139	0.268	0.216
8	5	55.441	30.692	0.990	2.863	65.068	8.384	0.224	0.179
8	5	49.370	31.196	0.993	2.511	60.174	9.558	0.290	0.216
8	5	36.540	32.503	0.995	2.341	57.285	10.252	0.231	0.183

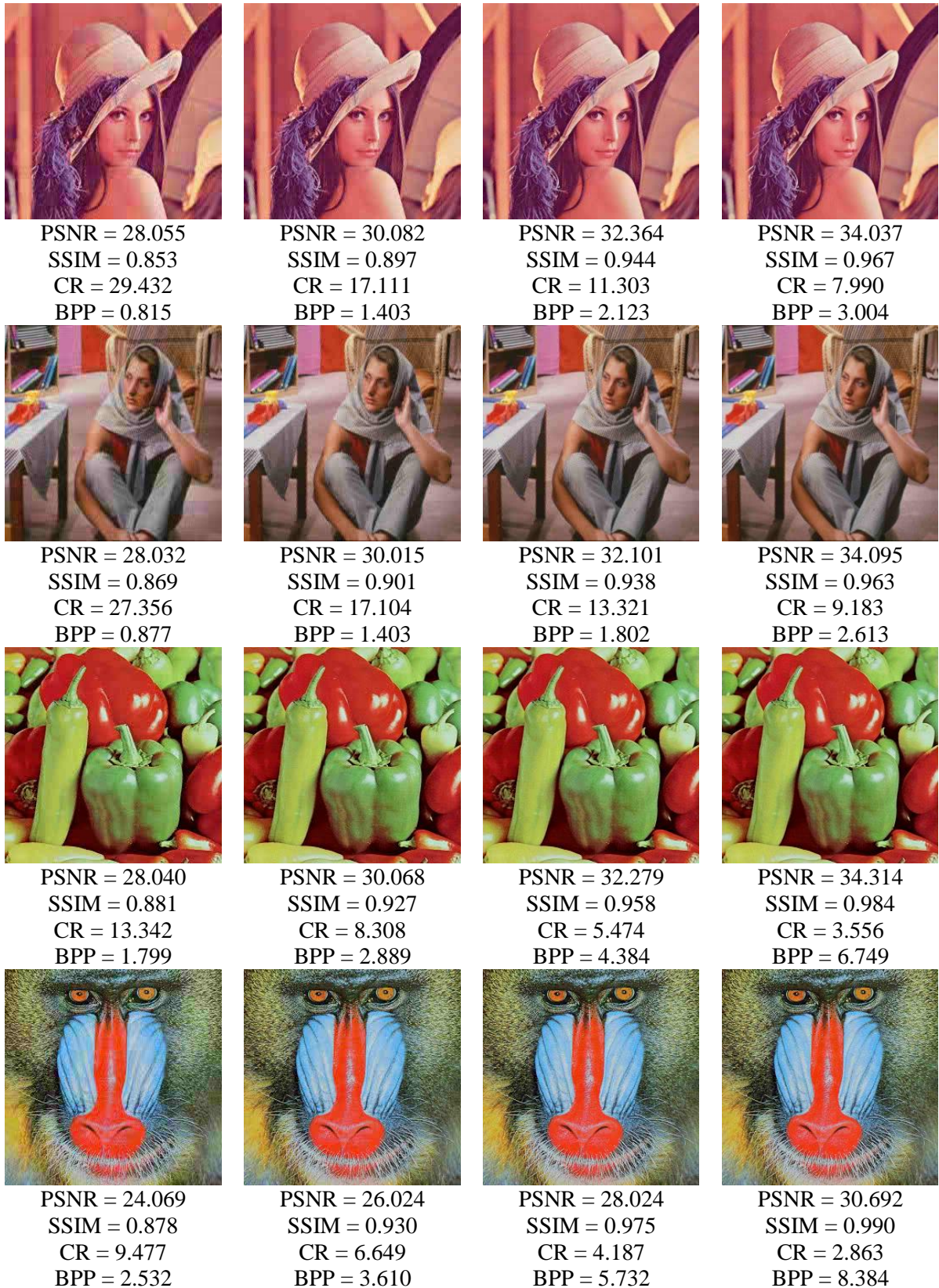


Figure 6- Samples of the reconstructed color images where the PSNR value is varied

8. Comparisons with JPEG standard

Tables 6 show the compression results (in terms of PSNR, SSIM, CR and BPP) which are attained by the proposed scheme with those given in standard encoder when they applied on color Lena, color Barbara, color Peppers, and color Baboon images, respectively; taking into consideration that different

image quality has been used. The listed results demonstrate the proposed systems compared with ACDsee Ultimate 2020 software results.

Table 6- Comparisons of the proposed system results with standard JPEG encoder

Test Image	Method	Size	PSNR	SSIM	CR	BPP
Color Lena	JPEG Standard	512×512	32.650	0.980	30.05	0.798
	Proposed Scheme	512×512	30.229	0.950	30.538	0.785
Color Barbara	JPEG Standard	256×256	31.507	0.939	17.389	1.380
	Proposed Scheme	256×256	30.015	0.901	17.104	1.403
Color Peppers	JPEG Standard	512×512	29.528	0.974	28.588	0.839
	Proposed Scheme	512×512	27.681	0.939	28.557	0.840
Color Baboon	JPEG Standard	512×512	24.403	0.978	15.332	1.565
	Proposed Scheme	512×512	23.371	0.937	15.086	1.5

9. Conclusions

In this paper, a new image compression module based on DCT was implemented. The proposed system was tested on several standard bitmap image files with $YCbCr$ color space to evaluate the system performance. The aim of the proposed system is to increase the CR while maintaining the PSNR of the reconstructed image as good as possible. From the obtained test results it can be notice that the threshold value that controls the level of quantization for each block in DCT encoder will vary according to the features of the block, whether it is a high frequency or low frequency blocks, which make the system more adaptive to deal with image data. The proposed entropy encoder has a positive significant impact on the results, it is important that the data of image must be processed in way to be suitable for coding process in order to make shift coding technique works efficiently to encode the input data. As a summary the best attained compression gains for the test images (Lena, Barbara, Peppers, and Baboon) are: 96.60%, 96.34%, 94.53%, 90.88% respectively, where PSNR value in an intermediate range.

Reference

1. Kaushik, A. and Gupta, M. **2012**. "Analysis of Image Compression Algorithms," *International Journal of Engineering Research and Applications*, pp. 773-779.
2. Luís, Eduardo, Sérgio, Nuno and Carla, **2017**. *Efficient Predictive Algorithms for Image Compression*, Springer International Publishing Switzerland.
3. Naik, H. **2015**. "A Review on Image Compression using Different Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, **5**(2): 181-184.
4. Nivedita, P. Singh and S. Jindal, **2012**. "A Comparative Study of DCT and DWT-SPIHT," *International Journal of Computational Engineering & Management*, **15**(2): 26-32.
5. Hashim, A.T. and Ali, S.A. **2016**. "Color Image Compression Using DPCM with DCT, DWT and Quadtree Coding Scheme," *Eng. &Tech. Journal*, **34**: 585-597, 2016.
6. Zhou, Y., Wang, C. and Zhou, X. **2018**. "DCT-Based Color Image Compression Algorithm Using an Efficient Lossless Encoder," *Image and Video Processing - Springer*, pp. 450-454.
7. Messaoudi, A., Benchabane, F. and Srairi, K. **2019**. "DCT-based color image compression algorithm using adaptive block scanning," *Signal, Image and Video Processing - Springer*, 2019.
8. Siddeq, M.M. and Rodrigues, M.A. **2017**. "A novel high-frequency encoding algorithm for image compression," *EURASIP Journal on Advances in Signal Processing*, pp. 1-17, 2017.
9. Dagher, I., Saliba, M. and Farah, R. **2018**. "Combined DCT-Haar transforms for image compression," *WILEY*, pp. 1-21.
10. ITU-R. **2011**. "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," *International Telecommunication Union*, 2011.

11. George, L.E., Hassan, E.K., Mohammed, S.G. and Mohammed, F.G. **2020**. "Selective Image Encryption Based on DCT, Hybrid Shift Coding and Randomly Generated Secret Key," *Iraqi Journal of Science*, **61**(4): 920-935.
12. Watson, A.B. **1984**. "Image Compression Using the Discrete Cosine Transform," NASA Ames Research Center, Mountain View, California.
13. Hassan, E.K., George, L.E. And Mohammed, F.G. **2018**. "Color Image Compression Based On Dct, Differential Pulse Coding Modulation, And Adaptive Shift Coding," *Journal of Theoretical and Applied Information Technology*, pp. 3160-3171.
14. Ahmed, S.D., George, L.E. and Dhannoon, B.N. **2015**. "The Use of Cubic Bezier Interpolation, Biorthogonal Wavelet and Quadtree Coding to Compress Color Images," *British Journal of Applied Science & Technology*, **11**(4): 1-11.
15. Ahmed, S.D., George, L.E. and Dhannoon, B.N. **2016**. "Image Compression Based on Cubic Bezier Interpolation, Wavelet Transform, Polynomial Approximation, Quadtree Coding and High Order Shift Encoding," *Research Journal of Applied Sciences, Engineering and Technology*, **13** (9): 696-705, 2016.
16. Anandan, R.S.S.P. **2016**. "Medical Image Compression Using Wrapping Based Fast Discrete Curvelet Transform and Arithmetic Coding," *Circuits and Systems*, pp. 2059-2069, 2016.
17. varnan, C., Jagan, A., Kaur,J., Jyoti, D. and Dr.D.S.Rao, D. **2011**. "Image Quality Assessment Techniques pn Spatial Domain," *International Journal of Computer Science and Technology*, **2**(3): 177-184, 2011.
18. Silpa, K. and Mastani, S. **2012**. "Comparison Of Image Quality Metrics," *International Journal of Engineering Research & Technology (IJERT)*, **1**(4): 1-5, 2012.
19. Nigam, A.K., Khare, P. and Srivastava, V.K. **2020**. "Image Compression Using Hybrid Approach and Adaptive Scanning for Color Images," *Advances in VLSI, Communication, and Signal Processing*, **587**: 761-774.