

Color image compression based on spatial and magnitude signal decomposition

Bushra A. Sultan¹, Loay E. George²

¹Department of Computer Science, University of Baghdad, College of Science, Baghdad, Iraq

²University of Information Technology and Communications, Baghdad, Iraq

Article Info

Article history:

Received Dec 21, 2020

Revised Mar 3, 2021

Accepted Mar 15, 2021

Keywords:

Bit plane

DCT

Delta PCM

Image compression

PSNR

ABSTRACT

In this paper, a simple color image compression system has been proposed using image signal decomposition. Where, the RGB image color band is converted to the less correlated YUV color model and the pixel value (magnitude) in each band is decomposed into 2-values; most and least significant. According to the importance of the most significant value (MSV) that influenced by any simply modification happened, an adaptive lossless image compression system is proposed using bit plane (BP) slicing, delta pulse code modulation (Delta PCM), adaptive quadtree (QT) partitioning followed by an adaptive shift encoder. On the other hand, a lossy compression system is introduced to handle the least significant value (LSV), it is based on an adaptive, error bounded coding system, and it uses the DCT compression scheme. The performance of the developed compression system was analyzed and compared with those attained from the universal standard JPEG, and the results of applying the proposed system indicated its performance is comparable or better than that of the JPEG standards.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Bushra A. Sultan

Department of Computer Science

College of Science, University of Baghdad

Baghdad, Iraq

Email: bushra542001@gmail.com, bushra.sultan@sc.uobaghdad.edu.iq

1. INTRODUCTION

Compression is a key mechanism applied in signal processing and has large significance because huge amounts of data are commonly transferred over a communication channel of a network [1]. Image compression (IC) is one of the techniques lay under image processing. It has many applications and plays an important role in the efficient transmission and storage of images. IC is a method used for reducing the size of the digital image in order to minimize the amount of space required to store it [1]. The two fundamental elements of compression are redundancy and irrelevancy reduction. Redundancy reduction aims to remove the duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver namely the human visual system (HVS) [2], [3]. According to the differences between the original image (uncompressed) and the reconstructed (decompressed) image, image compression is classified into lossless, exact retrieval, and lossy, approximate retrieval [4]-[7]. Signal decomposition is the extraction and separation of signal components from composite signals, which should preferably be related to semantic units. One important role of signal decomposition is improving the performance of compression algorithms [8]. IC techniques still experiencing many improvements under the discipline of transform coding. The transform coding schemes achieve higher compression ratios for lossy compression, but it suffers from some artifacts at high-compression ratios [9]. Discrete cosine transform

(DCT) is considered an important transformation used in data compression technology [10], it had an advantage in potential smaller dimension, better processing time, and compatibility with encoded data [11], its selection as the standard for joint photographic experts group (JPEG) is one of the major reasons for its popularity [12]. DCT helps to separate the image parts with respect to the image visual quality (i.e., high, low & middle-frequency components) [11], [13]. JPEG 2000 is an image coding algorithm; it is a modified variant of JPEG that uses a discrete wavelet transform (DWT) instead of DCT [10].

The main application of wavelet theory lies in the design of filters for sub-band coding. The basic concept behind the wavelet transform is the hierarchical decomposition of input image signal into a series of successively lower resolution reference signals and their associated detail signals [14]. Bit plane (BP) slicing is a simple and fast technique that highlights the contribution made by a specific bit and each biplane is a binary image [15]. It is a separations technique in which the image is sliced into different binary planes or layers according to bit position that efficiently analyzing the relative importance played by each bit of the image [16], [17], Al-Mahmood and Al-Rubaye [16], combined between bit plane slicing and adaptive predictive coding for lossless compression of natural and medical images, they utilized the spatial domain efficiently after discarding the lowest order bits namely, exploiting only the highest order bits in which the most significant bit corresponds to the last layer7 used adaptive predictive coding, while the other layers used run-length coding. Albahadily *et al.* [18], presented lossless grayscale image compression using the bit plane technique and modified run-length encoding (RLE). Al-Timimi [19] introduced a hybrid technique for lossless image compression of natural and medical images; it is based on integrating the bit plane slicing and Wavelet transform along with a mixed polynomial of the linear and nonlinear base. Many research groups have developed different image-coding schemes and tried to modify these schemes to further reduce bit rate [20].

The main problem is the need for reducing the size of the digital image in order to minimize; i) the amount of space required to store it, ii) the cost and time required to transfer it over the internet. Generally, spatial domain refers to the aggregate of pixels composing an image, so a digital color image consists of 3-spatial bands (red, green, and blue), each band pixel is represented using 8-bits (1 byte). Therefore, this paper's target is to improve the image compression system using spatial signal magnitude (value) decomposition and transform coding supported by an efficient entropy encoder. Where each image pixels decomposed into 2-values; most and least significant. According to the importance of the most significant value (MSV) that is influenced by any simple modification happened, an adaptive lossless image compression system has been proposed based on BP slicing, delta pulse code modulation, adaptive quadtree (QT) partitioning followed by an adaptive shift encoder. On another hand, the lossy compression system has been designed to handle the least significant value (LSV) based on an adaptive, error bounded, DCT based compression system. So, that the main contribution of this work is to improve an image compression system that utilizes the signal decomposition, where the proposed system is composed of two subsystems: lossless that compressed the MSV and lossy that compressed the LSV; the rest of the paper is structured as: Section 2 is dedicated to explaining the proposed compression system and method. The established proposed system is tested using some commonly used images, and the test results are discussed in section 3, finally, section 4 concludes the contribution of this paper.

2. THE PROPOSED SYSTEM AND METHOD

The proposed compression system consists of three main consecutive steps; where the layout of the proposed compression system illustrated in Figure 1:

- Step1 (load image): The uncompressed original BMP image is loaded and the three-color band arrays red (R), green (G), and blue (B) is extracted from it.
- Step2 (color transform): The digital image is converted from correlated RGB color system into less correlated color space components (like YUV) and produces 3 band arrays Y, U, and V.
- Step3 (value decomposition): Each value (magnitude) from the (Y, U, and V) bands is decomposed into 2-values LSV and MSV. The MSVs of each band are compressed lossless MSV encoder while the LSVs of each band are compressed lossy LSV encoder. The value of each color component is decomposed into two components (MSV stands for the most significant component, and LSV stands for the least significant number) using (1) and (2):

$$Band_{MSV}(x, y) = round\left(\frac{Band(x,y)}{Stp}\right) \quad (1)$$

$$Band_{LSV}(x, y) = Band(x, y) - Band_{MSV}(x, y) \times Stp \quad (2)$$

Where:

- $Band$ = is referred to the one of the less correlated color band (Y, U and V).
- x, y = is the row and column index of the band value
- Stp = is the value of the cut point bit. It is very important parameters that partitioned the values range [30..150]

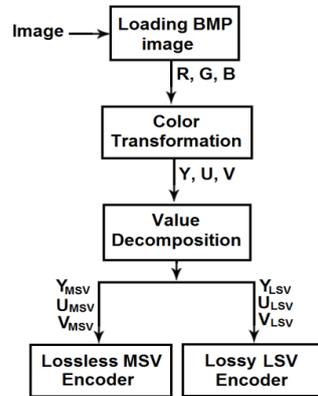


Figure 1. The proposed compression system layout

2.1. Lossless MSV encoder (LMSVE)

In this section, LMSVE has received the MSV 2-dimensional arrays for Y, U, and V bands, as illustrated in Figure 2. LMSVE consists of the following processes:

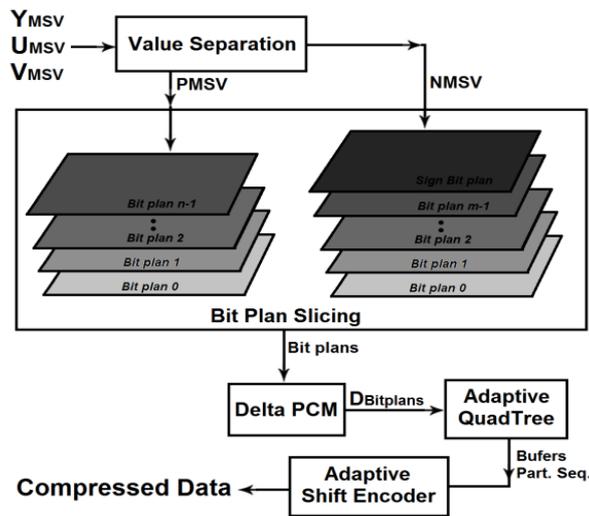


Figure 2. The lossless MSV encoder (LMSVE) diagram

2.1.1. Value separation

The task of this process is to isolate MSV into two positive and negative matrices, PMSV and NMSV respectively, to facilitate the coding process, in case of negative values occurrences NMSV will be discarded. The constructed matrices will be passed to the next process (BP slicing).

2.1.2. Bit plan slicing (BPS)

The minimum number of bits required to code 2^n distinct quantities is n, each bit represented in a layer so there are n-1 bit layers to represent 2^n positive distinct quantities. While the adopted layered representation of the negative -2^m was m bit layers the m layer represent the sign bit. The first step applied in this process is the number base conversion, so the decimal positive or negative number converted to its

binary equivalent. Later each bit is split to its corresponding layer number. When the layer is empty from 1's (all the content is 0's), it is marked as an empty to avoid encoding and gain storage space.

2.1.3. Delta pulse code modulation (Delta PCM)

Delta PCM is a special case from differential pulse code modulation applied to the sequence that consists of 0 or 1 values. It detects a change in the value from zero to one and vice versa (edges). The benefits of this process are to reduce the crowded bit plane and to make the bit plane closer to a sparse matrix. That increases the empty regions and facilitates the quadtree partitioning in the next process. The direction of calculating the delta must be zig-zag (inverted from line-to-line) as shown in Figure 3. Figure 3 (a) illustrates the direction, while Figure 3(b) explains the way of calculating the delta PCM for 3x3 window. So, Delta PCM will be applied to all non-blank plans that go through this process, in addition to counting the number of 1's in each resulted delta bitplane.

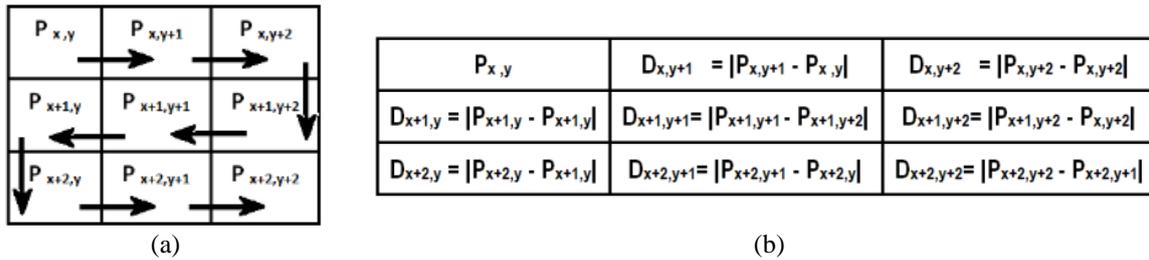


Figure 3. Delta PCM directions and functions, (a) The directions, (b) Delta PCM

2.1.4. Adaptive quad tree

It is well known that the purpose of using a quadtree with compression is to isolate empty areas in digital images and avoid encoding and storing them. To be a useful QT it must start with a block size appropriate to the size of the empty areas in the image, to avoid continuous division in the case of small areas or miss them in the case of large areas. Equations (3) and (4) are used to regulate the initial block size selected for the QT partitioning. The regulation is done using the rate of 1's that occurred in the entered delta bitplane see equations. The inputs to this process are delta bitplanes corresponding with each one the sum of all non-zero values (count the number of 1's) resulted from the previous process. The output from this process is the buffer of non-empty (2x2) blocks (i.e. series of 1 or 0 bits) beside the QT partitioning sequence for each entered delta bitplane.

$$Rate = \frac{1}{hw} \sum_{x=0}^{h-1} \sum_{y=0}^{w-1} Delta_{Bitplan}(x, y) \tag{3}$$

$$B_S = \begin{cases} 64 & \text{if } Rate \leq \alpha \\ 32 & \text{if } Rate \leq \alpha + inc \\ 16 & \text{if } Rate \leq \alpha + inc \times 2 \\ 8 & \text{if } Rate \leq \alpha + inc \times 3 \\ 4 & \text{Otherwise} \end{cases} \tag{4}$$

Where:

- $Delta_{Bitplan}$ = Referred to the entered delta bitplane
- x, y = is the row and column index of the delta bitplane respectively
- h, w = is the height and width of the delta bitplane respectively
- B_S = is the Initial block size used in QT
- inc = is real increment {0.2,0.3}

2.1.5. Adaptive shift encoder

This process is responsible for performing the final step of the lossless MSV encoder (LMSVE), by encoding the resulted data from the previous step (buffers and partitioning sequences). An adaptive shift optimizer is the first process that must perform before applying the shift encoding. The following steps summarize the shift coding optimizer and an adaptive shift coding process (steps 1, 2, 3 belongs to the shift coding optimizer):

- Step 1: convert each adjacent 4-bits from the buffer to its decimal equivalents and this process is applied to all bits in the buffer and store the results in a 1-d integer number entitled $NUM()$, so the range of the value in the buffer is [1]-[15].
- Step 2: compute the histogram for $NUM()$, and apply the histogram packing, so the new histogram includes only the non-zero occurrences number entitled $Ph()$.
- Step 3: compute the following equations:

$$N_{02} = Ph(0) + Ph(1) + Ph(2) \quad (5)$$

$$N_{35} = Ph(3) + Ph(4) + Ph(5) \quad (6)$$

$$N_{68} = Ph(6) + Ph(7) + Ph(8) \quad (7)$$

$$N_{69} = N_{68} + Ph(9) \quad (8)$$

$$N_{CE} = Ph(12) + Ph(13) + Ph(14) \quad (9)$$

$$N_{9E} = Ph(9) + Ph(10) + Ph(11) + N_{CE} \quad (10)$$

$$N_{AE} = Ph(10) + Ph(11) + N_{CE} \quad (11)$$

Compute the total required bits (Tb) for the following possible shift coding combinations:

$$Tb_0 = 4 \times (N_{02} + N_{35} + N_{69} + N_{AE}) \quad (12)$$

$$Tb_1 = 2 \times N_{02} + 4 \times N_{35} + 6 \times N_{68} + 9 \times N_{9E} \quad (13)$$

$$Tb_2 = 2 \times N_{02} + 5 \times (N_{35} + N_{69}) + 8 \times N_{AE} \quad (14)$$

$$Tb_3 = 2 \times N_{02} + 6 \times (N_{35} + N_{69} + N_{AE}) \quad (15)$$

- Step 4: according to the computed (Tb_0 , Tb_1 , Tb_2 , and Tb_3) values the implemented four shifts coding schemes (Sh_c) had been used to store the data as listed in the following relation:

$$Sh_c_i = \begin{cases} i = 0 & \text{if } \min(Tb_0, Tb_1, Tb_2, Tb_3) = Tb_0 \\ i = 1 & \text{if } \min(Tb_0, Tb_1, Tb_2, Tb_3) = Tb_1 \\ i = 2 & \text{if } \min(Tb_0, Tb_1, Tb_2, Tb_3) = Tb_2 \\ i = 3 & \text{if } \min(Tb_0, Tb_1, Tb_2, Tb_3) = Tb_3 \end{cases} \quad (16)$$

The number of bits NB (codeword length) required to store each value from $NUM()$ is determined according to the following equations which are all relative to the sequence of shift coding:

$$NB_{Sh_c0} = 4 \forall Val \quad (17)$$

$$NB_{Sh_c1} = \begin{cases} 2 & \text{if } Val < 3 \\ 2,2 & \text{if } Val < 6 \\ 2,2,2 & \text{if } Val < 9 \\ 2,2,2,3 & \text{Otherwise} \end{cases} \quad (18)$$

$$NB_{Sh_c2} = \begin{cases} 2 & \text{if } Val < 3 \\ 2,3 & \text{if } Val < 10 \\ 2,3,3 & \text{Otherwise} \end{cases} \quad (19)$$

$$NB_{Sh_c3} = \begin{cases} 2 & \text{if } Val < 3 \\ 2,4 & \text{Otherwise} \end{cases} \quad (20)$$

Where Val is the integer value stored in $NUM()$. For more illustration about the behaviour of the four shifts coding schemes adopted in this paper, Figure 4 selected (18) to determine how many bits need to encode each Val, where NB is the number of bits required. The steps from 1 to 4 are applied on the buffer but the QT partitioning sequence compressed using traditional shift coding after collecting each adjacent 8-bits.

2.2. Lossy LSV encoder (LLSVE)

In this section, LLSVE has received the LSV 2-dimensional arrays for Y, U, and V bands, as illustrated in Figure 5. LLSVE consists of the following processes:

2.2.1. Band partitioning

This process is responsible for dividing the received LSV 2-D array into non overlapped blocks of size n (BLK nxn), to perform the following processes on blocks level.

2.2.2. DCT transform

For all block entry apply the well-known DCT transform [21], to separate the block into parts with respect to the importance i.e., high, low & middle-frequency components.

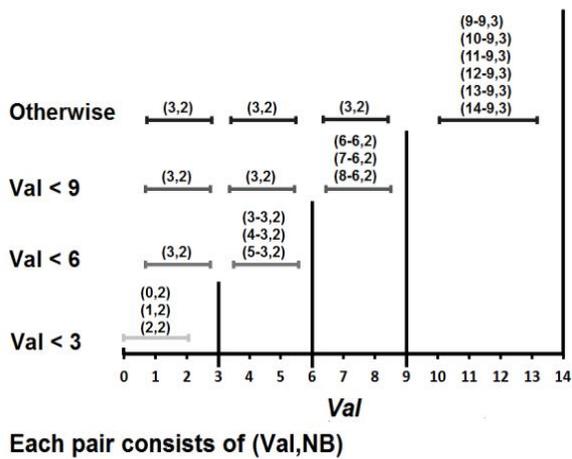


Figure 4. An illustration for (18)

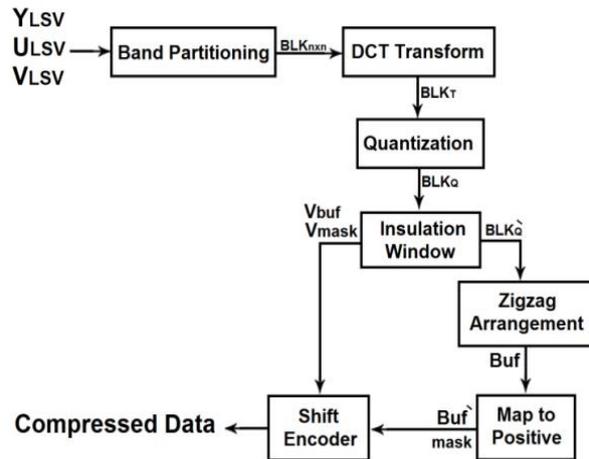


Figure 5. The lossy LSV encoder (LLSVE) diagram

2.2.3. Quantization

This process is responsible for increase the compression gain by eliminating insignificant data. Due to the distinct importance of the DCT coefficients, ununiformed scaler quantization had been applied according by (21), (22) [22]:

$$BLK_Q(i, j) = round\left(\frac{BLK_T(i+u, j+v)}{Q_{stp}(u, v)}\right) \text{ for } u \& v \in [0, n - 1] \tag{21}$$

$$Q_{stp}(u, v) = \begin{cases} q & \text{for } u = v = 0 \\ q_0(1 + \alpha(u, v)) & \text{otherwise} \end{cases} \tag{22}$$

Where:

- BLK_T = is the DCT coefficients of the block indexed i, j
- BLK_Q = is the quantized DCT coefficients of the block indexed i, j
- i, j = is the upper left corner index of the block, started at row i, and column j
- n = is the size of the block
- Q_{stp} = is the quantization matrix relative to each DCT block coefficients
- q = is the quantization value of the DC coefficient that set to $2n$
- q_0 = is the quantization value of the AC coefficient $\in \{2,3\}$
- α = is the scaling factor $\in \{0.1, 0.12, \dots, 0.2\}$

2.2.4. Insulation window

It is known that the large and effective values of the DCT coefficients' are centered in the northern left quarter of the block, so to improve the shift encoder and increase the compression gain, the large value coefficients will be moved to their own buffer. This process is done by opening a window with a size of (m x m), m is an integer ranged [10..16] relative to the block size (n), that started in the upper left corner of the block and checks each quantized DCT coefficient's if they are greater than S (i.e., a predefined value)

they will be isolated to buffer. All the quantized DCT coefficient blocks will be passed through the insulation window according to algorithm (1) steps in Figure 6.

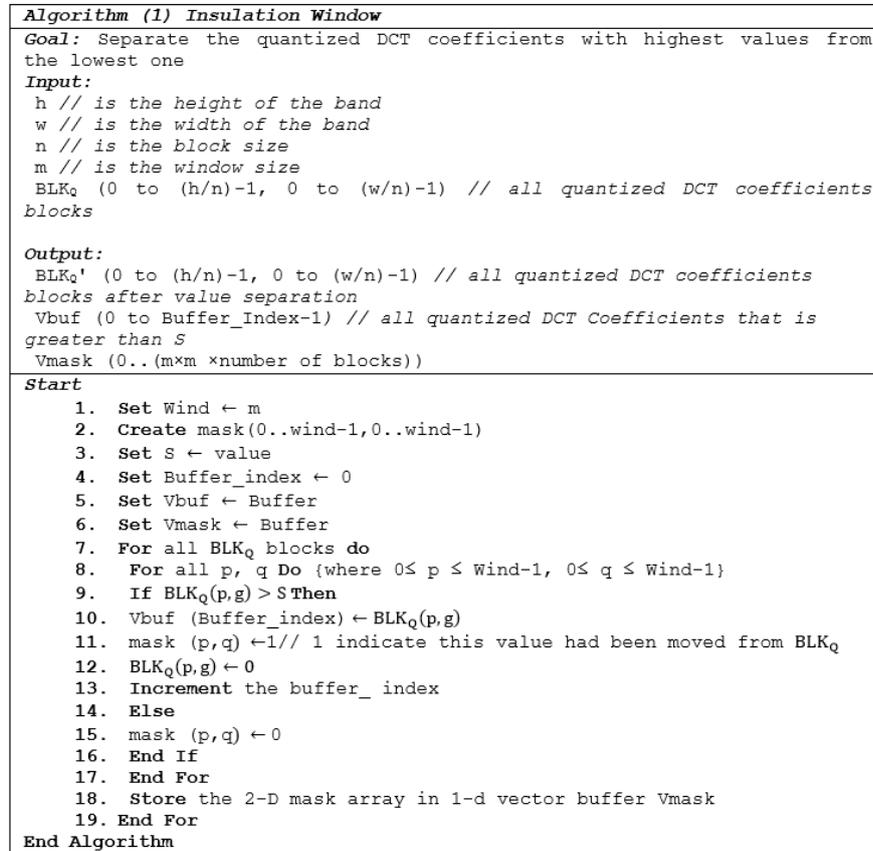


Figure 6. Algorithm of insulation window

2.2.5. Zigzag arrangement

This process is applied to convert the DCT coefficient blocks from 2-d arrays to one dimension buffer, to increase the benefit of spatial correlation as illustrated in Figure 7 [23].

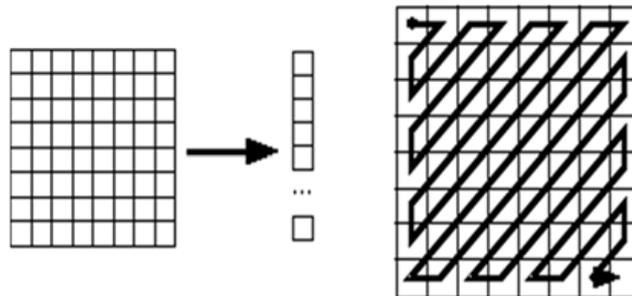


Figure 7. Zigzag arrangement illustration [23]

2.2.6. Map to positive

The 1-d buffer of all the quantized DCT coefficients' after separating the highest values from it will pass to this process, which responsible for converting all the numbers to positive to avoid coding complexity according by (23) [24, 25]:

$$X_i = \begin{cases} 2X_i & \text{if } X_i \geq 0 \\ -2X_i - 1 & \text{if } X_i < 0 \end{cases} \quad (23)$$

Where, X_i is the i^{th} element value registered in the buffer.

2.2.7. Shift encoder

This process will receive the three buffers: $V_{buf}()$ which contains the highest quantized DCT coefficients' with their masks buffer ($V_{mask}()$) that's need to recombine the values and the lowest quantized DCT coefficients' after converted to positive ($buf''()$). Due to that separation, the shift encoder will process each buffer separately to increase the compression gain and the output for each buffer will be appended to the compressed data.

2.3. Decoding system

This system is responsible to reconstruct the decompressed image. The compressed data from LLSVS appended to that of LMSVS will be passed to a decoding system that consisted of two sub decoding systems: the decoding subsystem of the LMSVS compressed data and the decoding subsystem of the LLSVS compressed data. Each decoding subsystem consists of the same processes proposed in the encoding subsystem but reverse order. So, the resulted data from the two decoding subsystems will be recombined to reconstruct the new Y, U, and V bands that will be transformed back to the RGB color model to extract the decompressed image.

3. RESULTS AND DISCUSSION

The proposed system was implemented using Embarcadero RAD Studio 2010 (Delphi 2010) programming language. The tests were conducted under the environment: Windows-8 operating system, laptop computer Lenovo (processor: Intel (R) Core(TM) i5-3337U, CPU 1.8 GHz and 4 GB RAM). The tests were applied on the well-known Lena and Baboon image samples (whose specifications are: size=256x256 pixel, color depth=24 bit), as shown in Figure 8. In an image compression scheme, the image compression algorithm should attain a trade-off between compression ratio and image quality [26], so to assess the difference between the reconstructed image and the original images, the error measure (i.e. peak signal to noise ratio PSNR measured in dB) was used. Beside this fidelity measure, some complementary measure was used to describe the performance of the system; the compression ratio (CR) was used to describe the compression gain. Table 1 lists the considered control parameters (including their names and default values); these values were selected after making comprehensive tests. Table 2 illustrates the effect of Stp on the number of bytes required to encode the LMSVE (NB_M) for all tested images.



Figure 8. Selected image samples for tests; (a) Lena and (b) Baboon

Table 1. The values of the control parameters

Notation	Range	Scope of Effects
Stp	{30,40, ...,150}	LMSVE & LLSVE
inc	0.2	LMSVE
n	{32,64,128}	LLSVE
q	$2 \times n$	LLSVE
q_0	{2,3}	LLSVE
α	{0.1,0.12,0.14, ...,0.2}	LLSVE
S	7	LLSVE
m	10	LLSVE

Table 2 shows that the increase of Stp value will relatively decrease NB_M in most cases. Table 3 illustrates the effect of Stp, q_0 and α on the number of bytes required to encode the LLSVE (NB_L) for the Baboon image when $n=64$. The Stp effects on NB_M listed in Table 2 are used with NB_L listed in Table 3 to compute the CR of the proposed compression system. Table 3 proved that the increase of the quantization's factors (q_0 and a) will case increasing the CR and decreasing the PSNR of the retrieved image. Table 4 illustrates the effect of (n , Stp and α) on CR and PSNR, all tests applied on Lena image when ($q_0 = 2$).

Table 2. The number of bytes required to encode the LMSVE

Stp	NB_M		Stp	NB_M	
	Lena	Baboon		Lena	Baboon
30	24514	32100	95	9525	11519
35	24868	29189	100	9620	10448
40	18677	26546	105	9398	9706
45	14986	25113	110	8144	9009
50	12980	22744	115	6983	7932
55	12158	18204	120	5553	6476
60	11528	16168	125	4482	5027
65	10378	15015	130	4086	4014
70	9967	14707	135	3624	3571
75	9161	14232	140	3232	3742
80	8967	13443	145	2672	3870
85	9212	13032	150	2332	4015
90	8999	12395			

Table 3. The effect of (Stp, q_0 and α) on the fidelity measurements for Lena test image when $n=64$

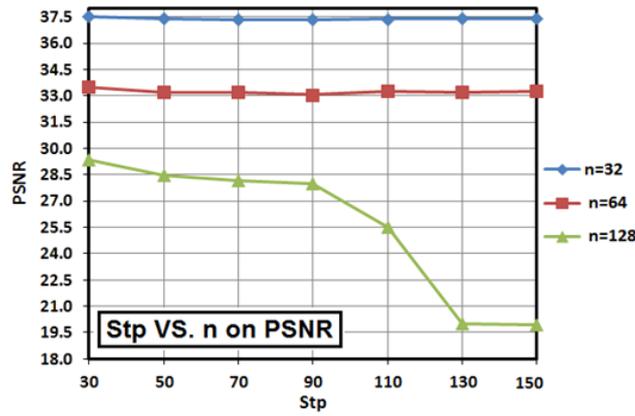
Stp	α	PSNR	$q_0=2$		$q_0=3$		
			NB_L	CR	PSNR	NB_L	CR
30	0.10	33.64	5239	6.61	31.08	4621	6.75
	0.12	32.58	5022	6.66	30.23	4567	6.76
	0.14	31.74	4826	6.70	29.58	4345	6.81
	0.16	31.00	4899	6.68	29.08	4112	6.87
	0.18	30.44	4810	6.70	28.65	4012	6.89
	0.20	29.98	4700	6.73	28.32	3813	6.94
50	0.10	33.34	5886	10.42	30.71	5399	10.70
	0.12	32.26	5762	10.49	29.77	5166	10.83
	0.14	31.36	5596	10.58	29.06	4988	10.94
	0.16	30.62	5519	10.63	28.47	4865	11.02
	0.18	29.99	5471	10.66	27.97	4775	11.07
	0.20	29.48	5423	10.68	27.56	4592	11.19
70	0.10	33.32	6232	12.14	30.66	5787	12.48
	0.12	32.26	6168	12.19	29.68	5484	12.72
	0.14	31.33	6122	12.22	28.88	5485	12.72
	0.16	30.57	6097	12.24	28.22	5433	12.77
	0.18	29.91	5778	12.49	27.67	5239	12.93
	0.20	29.33	5864	12.42	27.21	5182	12.98
90	0.10	33.39	6534	12.66	30.64	6063	13.05
	0.12	32.30	6352	12.81	29.58	5974	13.13
	0.14	31.37	6300	12.85	28.71	5760	13.32
	0.16	30.53	6243	12.90	27.94	5598	13.47
	0.18	29.82	6180	12.95	27.30	5542	13.52
	0.20	29.21	6140	12.99	26.73	5524	13.54
110	0.10	33.44	6572	13.36	30.60	6018	13.88
	0.12	32.31	6513	13.41	29.51	6028	13.87
	0.14	31.33	6473	13.45	28.61	4826	15.16
	0.16	30.50	6425	13.50	27.84	5800	14.10
	0.18	29.77	6265	13.64	27.19	5708	14.19
	0.20	29.12	6228	13.68	26.60	5556	14.35
130	0.10	33.48	6269	18.99	30.73	5687	20.12
	0.12	32.39	6214	19.09	29.67	5684	20.12
	0.14	31.42	6137	19.23	28.82	5472	20.57
	0.16	30.63	5979	19.53	28.10	4419	23.12
	0.18	29.92	5913	19.66	27.47	5244	21.07
	0.20	29.30	5880	19.73	26.92	5189	21.20
150	0.10	33.43	6302	22.77	30.78	5673	24.56
	0.12	32.38	6208	23.02	29.78	5738	24.36
	0.14	31.48	6139	23.21	28.96	5406	25.41
	0.16	30.69	6088	23.35	28.25	5307	25.74
	0.18	30.03	5942	23.76	27.63	5268	25.87
	0.20	29.43	5754	24.31	27.11	5207	26.08

Table 4. The effect of (n, Stp and α) on the fidelity measurements for Baboon test image when $q_0=2$

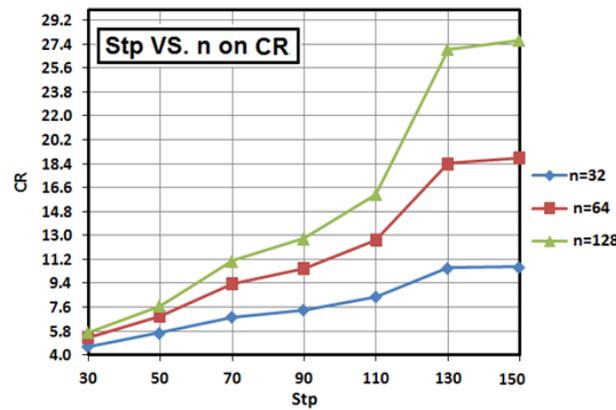
Stp	α	n=32		n=64		n=128	
		PSNR	CR	PSNR	CR	PSNR	CR
30	0.10	37.51	4.61	33.49	5.32	29.37	5.71
	0.12	36.58	4.65	32.37	5.31	28.56	5.69
	0.14	35.75	4.70	31.43	5.36	27.98	5.73
	0.16	34.97	4.74	30.67	5.39	27.55	5.73
	0.18	34.25	4.76	30.02	5.40	27.23	5.75
	0.20	33.62	4.81	29.49	5.39	26.98	5.76
50	0.10	37.39	5.67	33.20	6.90	28.46	7.68
	0.12	36.45	5.57	32.01	6.96	27.31	7.72
	0.14	35.56	5.66	30.97	6.97	26.43	7.75
	0.16	34.74	5.73	30.08	7.00	25.73	7.76
	0.18	34.02	5.78	29.31	7.03	25.18	7.77
	0.20	33.36	5.82	28.65	7.05	24.75	7.78
70	0.10	37.36	6.83	33.19	9.35	28.17	11.08
	0.12	36.40	6.93	31.96	9.37	26.93	11.09
	0.14	35.53	7.02	30.91	9.40	25.95	11.10
	0.16	34.73	7.11	29.97	9.49	25.16	11.12
	0.18	33.99	7.20	29.15	9.51	24.53	11.17
	0.20	33.34	7.32	28.44	9.63	24.01	11.24
90	0.10	37.33	7.39	33.04	10.51	28.00	12.76
	0.12	36.38	7.51	31.84	10.54	26.78	12.83
	0.14	35.53	7.63	30.82	10.58	25.81	12.85
	0.16	34.73	7.74	29.91	10.67	25.01	12.86
	0.18	34.00	7.84	29.10	10.69	24.35	12.88
	0.20	33.35	7.91	28.39	10.73	23.80	12.89
110	0.10	37.38	8.38	33.24	12.63	25.50	16.10
	0.12	36.44	8.52	32.05	12.68	27.10	16.12
	0.14	35.57	8.64	31.06	12.81	26.09	16.25
	0.16	34.81	8.80	30.18	13.04	25.25	16.31
	0.18	34.12	9.01	29.41	13.08	24.54	16.33
	0.20	33.47	9.13	28.70	13.14	23.96	16.47
130	0.10	37.40	10.55	33.21	18.41	20.00	27.03
	0.12	36.44	10.78	32.02	18.75	19.79	27.10
	0.14	35.63	10.98	31.00	19.05	19.58	27.17
	0.16	34.82	11.21	30.14	19.17	19.38	27.41
	0.18	34.12	11.56	29.35	19.27	20.92	27.46
	0.20	33.48	11.75	28.67	19.35	22.10	27.82
150	0.10	37.45	10.33	33.21	18.26	19.96	27.72
	0.12	36.53	10.57	32.07	18.54	25.13	26.61
	0.14	35.70	10.81	31.07	18.63	26.93	26.69
	0.16	34.95	11.03	30.18	18.92	25.89	26.77
	0.18	34.28	11.35	29.39	18.99	25.00	26.89
	0.20	33.65	11.54	28.70	19.17	24.25	27.25

Figure 9 shows the effect of (n and Stp) on (CR and PSNR) on the test image Baboon, where the test was done with ($q_0=2$ and $\alpha=0.01$,). From Figure 9, we conclude that when $n=32$ with different Stp values, the compression ratio is close to Lossless and the PSNR of the reconstructed image is higher than the acceptable levels (37.33-37.51), where it is effective (LMSVS) over from the effectiveness of (LLSVS). But when the value of n increases to become 64, Stp begins to become more active, affecting a noticeable increase in the compression ratio, with the slight decrease of the PSNR of the retrieved image remaining within very acceptable levels (33.04-33.49). Finally, when $n=128$, the influence of (LLSVS) begins to overwhelm the effect of ((LMSVS), causing an inverse relationship, with a clear difference, between the PSNR of the decompressed image and the CR, causing a significant increase in CR with a clear decrease in the PSNR, especially when Stp=(130 and 150).

From Figure 9, we conclude that when $n=32$ with different Stp values, the compression ratio is close to Lossless and the PSNR of the reconstructed image is higher than the acceptable levels (37.33-37.51), where it is effective (LMSVS) over from the effectiveness of (LLSVS). But when the value of n increases to become 64, Stp begins to become more active, affecting a noticeable increase in the compression ratio, with the slight decrease of the PSNR of the retrieved image remaining within very acceptable levels (33.04-33.49). Finally, when $n=128$, the influence of (LLSVS) begins to overwhelm the effect of (LMSVS), causing an inverse relationship, with a clear difference, between the PSNR of the decompressed image and the CR, causing a significant increase in CR with a clear decrease in the PSNR, especially when Stp=(130 and 150). The obtained results of the proposed compression system are compared to the universal standard JPEG regarding compression ratio and visual quality measures, Table 5 illustrates this comparison that applied to the Lena test image.



(a)



(b)

Figure 9. The effect of n and Stp, (a) Stp Vs. n on PSNR, (b) Stp Vs. n on CR

Table 5. The proposed compression system comparison with JPEG 2000 for the test image Lena

JPEG Standard			Proposed Method Parameter Setting					
Quality	CR	PSNR	Stp	n	q_0	α	CR	PSNR
80%	10.32	34.56	150	32	2	0.18	14.57	34.52
			140	32	2	0.18	13.66	34.53
			130	32	2	0.18	12.88	34.48
40%	21.59	31.6	115	32	2	0.16	10.04	35.03
			140	64	2	0.18	21.46	29.98
			140	64	2	0.20	21.74	29.36
20%	23.35	30.77	145	64	2	0.10	21.93	33.47
			145	64	2	0.12	22.20	32.42
			150	64	2	0.12	23.02	32.38
			145	64	2	0.20	23.19	29.42
			150	64	2	0.14	23.21	31.48
			150	64	2	0.16	23.35	30.69

4. CONCLUSION

From the results of tests conducted on the proposed system, the following remarks were stimulated: The system is progressive as taking into consideration the importance of the most significant pixel values (MSV) (MSV take a small number of bits after encoding, knowing that they are encoded without error while least significant pixel values (LSV) are handled using lossy coding as they require a large number of bits. The increase of the control parameter Stp (that is the cut point bit) will relatively decrease the number of bits required to encoding the MSV in most cases, as shown in Table 2. The control parameter n (that is the size of the block partitioning) was tested in lossless MSV compression sub-system (LLSVS) using non overlapped scheme is plays an important role in the proposed compression system. When (n=32) makes the system biased to the LMSVS, but when (n=128) the system becomes biased to LLSVS. While, when (n=64)

represented the balancing mode between the two subsystems (LMSV and LLSVS) that produce very encouraging values in term of CR and PSNR, as shown in Table 4 and Figure 9. For the future, it is possible to develop the Entropy encoder method to increase the compression ratio or using the multi-stage of transform.

REFERENCES

- [1] A. Kapoor and R. Dhir, "Image Compression Using Fast 2-D DCT Technique," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 6, pp. 2415-2419, Jun. 2011.
- [2] S. E. Umbaugh, "Digital Image Processing and analysis: Human and Computer Vision Applications with CVIPtools," *CRC Press*, 2nd Edition, 2011.
- [3] A. J. Qasim, R. Din, and F. Q. A. Alyousuf, "Review on techniques and file formats of image compression," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 2, pp. 602-610, Apr. 2020, doi: 10.11591/eei.v9i2.2085.
- [4] D. Salomon and G. Motta, "Data Compression the Complete Reference," *Springer*, 4th Edition, 2007.
- [5] Praveena P., "Implementation of LOCO-I Lossless Image Compression Algorithm for Deep Space Applications," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 3, no. 3, pp. 98-103, Nov. 2014, doi: 10.11591/ijres.v3.i1.pp25-30.
- [6] D. Narmadha, K. Gayati, Thilagavati, and N. S. Basha, "An Optimal HSI Image Compression using DWT and CP," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 3, pp. 411-421, Jun. 2014, doi: 10.11591/ijece.v4i3.6326.
- [7] E. Setyaningsih, and A. Harjoko, "Survey of Hybrid Image Compression Techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 4, pp. 2206-2214, Aug. 2017, doi: 10.11591/ijece.v7i4.pp2206-2214.
- [8] J. R. Ohm, "Multimedia Communication Technology," *Springer-Verlag*, Berlin, Heidelberg, 2004, doi: 10.1007/978-3-642-18750-6.
- [9] O. Egger, P. Fleury, T. Ebrahimi and M. Kunt, "High-performance compression of visual information-a tutorial review. I. Still pictures," in *Proceedings of the IEEE*, vol. 87, no. 6, pp. 976-1013, Jun. 1999, doi: 10.1109/5.763312.
- [10] A. Kurniawan, T. W. Purboyo, and A. L. Prasasti, "Implementation of Image Compression Using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT)," *International Journal of Applied Engineering Research*, vol. 12, no. 23, pp. 13951-13958, 2017.
- [11] R. A. R. Ahmad, M. I. Ahmad, M. N, M, Isa, and S. A. Anwar, "Face recognition using assemble of low frequency of DCT features," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 541-550, Jun. 2019, doi: 10.11591/eei.v8i2.1417.
- [12] A. M. Raid, W. M. Khedr, M.A. El-dosuky, and W. Ahmed, "Jpeg Image Compression Using Discrete Cosine Transform - A Survey," *International Journal of Computer Science and Engineering Survey (IJCSSES)*, vol. 5, no. 2, pp. 39-47, Apr. 2014, doi: 10.5121/ijcses.2014.5204.
- [13] S. Devi, and A. Kalia, "Image Compression Using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) Techniques," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 5, no. 10, pp. 1689-1696, Oct. 2017.
- [14] J. D. Villasenor, B. Belzer and J. Liao, "Wavelet filter evaluation for image compression," in *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1053-1060, Aug. 1995, doi: 10.1109/83.403412.
- [15] Makandar A. and Somshekar R., "Alzheimer's disease detection using bit plane slicing Images," *International Journal of Engineering Development and Research*, vol. 6, no. 3, pp. 405-411, 2018.
- [16] H. Al-Mahmood. and Z. Al-Rubaye, "Lossless Image Compression based on Predictive Coding and Bit Plane Slicing," *International Journal of Computer Applications*, vol. 93, no 1, pp. 1-6, May 2014, doi: 10.5120/16176-2068.
- [17] S. A. Abu Taleb, H. M. J. Musafa, A. M Khtoom, and I. K. Gharaybih, "Improving LZW Image Compression," *European Journal of Scientific Research*, vol. 44, no. 3, pp. 502-5093, 2010.
- [18] H. K. Albahadily, V. Y. Tsviatkou, and V. K. Kanapelka, "Grayscale Image Compression using Bit Plane Slicing and Developed RLE Algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 2, pp. 309-314, Feb. 2017.
- [19] R. T. Al-Timimi, "Bit Plane Slicing, Wavelet and Polynomials Mixing for Image Compression," *Iraqi Journal of Science*, vol. 60, no. 11, pp. 2497-2505, 2019.
- [20] Y. Wongsawat, H. Ochoa, K. R. Rao and S. Oraintara, "A Modified Hybrid DCT-SVD Image-Coding System for Color Image," *International Symposium on Communications and Information Technologies (ISCIT 2004) Sapporo, Japan*, 2004, pp. 766-769.
- [21] D. Salomon and G. I. Motta; "Handbook of Data Compression," *Springer*, 5th Edition, 2010.
- [22] E. Kh. H. Hassan, L. E. George, and F. G. Mohammed, "Color Image Compression Based On DCT, Differential Pulse Coding Modulation, and Adaptive Shift Coding," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 11, pp. 3160-3171, June 2018.
- [23] N. T. Rao and S. K. Devireddy, "Image Compression Using Discrete Cosine Transform," *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, vol. 17, no. 3, pp. 81-88, 2008.

- [24] L. E. George and B. A. Sultan, "The Use of Biorthogonal Wavelet, 2D Polynomial and Quadtree to Compress Color Images," *Springer-Verlag*, Berlin, Heidelberg, vol. 260, pp. 3790-390, 2011.
- [25] L. E. George and B. A. Sultan, "Image Compression Based On Wavelet, Polynomial and Quadtree," *Journal of Applied Computer Science & Mathematics*, vol. 11, no. 5, pp. 15-20, 2011.
- [26] A. Hazarathiah and R. B. Prabhakara, "Medical Image Compression using Lifting based New WaveletTransforms," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 5, pp. 741-750, 2014, doi: 10.11591/ijece.v4i5.5969.

BIOGRAPHIES OF AUTHORS



Bushra Abdullah Sultan had graduated from the University of Technology at 1995, had got MSc. From the University of Al-Nahrain at 2001 and PhD from the University of Technology at 2016. Currently she is a member of teaching staff in the department of computer science/college of science/University of Baghdad, Iraq. Her main research concerns are in: 1) Digital Multimedia Processing; 2) Coding Data (compression, representation).



Loay Edwar George had graduated from Baghdad University at 1979, had got MSc. at 1982 and PhD at 1997 in digital image processing. Currently, working as assistant of UoITc President for Scientific Affairs. His main research concerns are: 1) Digital Multimedia Processing; 2) Coding (encryption, digital signature, data compression, representation); 3) Pattern Recognition & Classification; 4) Fast Strings Processing and Analysis; 5) Biometrics; 6) Visual Based application.