Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 1261–1276
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.7013

Adaptive 1-D polynomial coding to compress color image with C421

Samara S. AL-Hadithy*, Ghadah K. AL-Khafaji

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

(Communicated by Javad Vahidi)

Abstract

Color images, in spite of it, are quick and simple to understand (interpret) by our minds, but digitally suffer from huge byte consumptions that directly affect storage (space) and/or transmission. Color data compression systems are vital to control the enormous size of color data by losing insignificant color data that implies redundancies and manipulating significant color information. This research is concerned with reducing or packing the storage of color visual natural images using color transformation of YC_bC_r base, based on a hybrid spatial-frequency compression technique. The essential part of the source band that proposed compression system constitutes of exploiting the adaptive lossy 1-D polynomial coding for coefficients and Minimize Matrix Size Algorithm (MMSA) for residual. On the other hand, for the non-source bands, the improved hierarchal scalar quantization method called double scalar uniform quantization scheme (DSUQS) is exploited along C421. For testing the performance of the suggested system, four natural color images from two datasets were adopted, the first called (Miscellaneous) and the second called (Kodak) of square sizes either of (512×512) pixels. The experimental results showed the superiority of YC_bC_r compared to the well-known standard joint photographic expert group (JPEG) that achieved higher performances where CR between (21-27) and PSNR between (36-38) dB, where CR of JPEG between (16-21) and PSNR (30-33) dB that implicitly means that the proposed system of color transformation base improved on average compared to JPEG with better-perceived quality for the compressed image.

Keywords: Color Image, YCbCr, MMSA, C421, DSUQS, Miscellaneous, Kodak
2020 MSC: 11H71, 68U10

1 Introduction

Today, with the marvelous rapidly evolving and changing in technology increasingly ubiquitous use of digital devices (i.e., computers, mobiles, tablets) and products (applications) (i.e., social media, electronic-learning, and instant messaging), where the number of people that are active online exceeds 2.5 billion [**?**], the color image or digital color image is intensively used, due to visual and emotional ability to convey the message (story) obviously, which exceed the television, cinema to encompass learning, marketing, electronic government (E-G), security, health care, and satellite fields [2].

*Corresponding author
Email addresses: samara.khalid1201@sc.uobaghdad.edu.iq (Samara S. AL-Hadithy), ghada.toma@sc.uobaghdad.edu.iq (Ghadah K. AL-Khafaji)

Received: August 2022 *Accepted:* October 2022

Where compression techniques become an urgent needs to manipulate data storage and transmission [3], the techniques varies according to the types and way of redundancy removal, where the former implies lossless, lossy and hybrid base techniques, while the latter implies spatial coding (SC), and /or transform coding (TC) techniques [7].

However, the necessity demand for adopting new color compression systems grew that led to new sophisticated scheme standard of JEPG 2000 (JP2), the question arose as to why most of the improved compression systems were concerned with JPEG rather than JP2 in spite of much better compression solution techniques, JPEG characterized by simplicity, fast and widely utilized by websites and camera manufacturers compared to the JP2 that characterized by complexity, slower and adopted by certain browses [9]. Currently time even with these tremendous standard performances, the need for innovation compression schemes become intensive research that is compatible with the capabilities of data type and needs.

This paper is concerned with compression natural color images that increasingly important applications with concurrent performance compared to the current standard techniques of JPEG base. The core approach taken is to try to investigate a hybrid color image coding system based on improved the modern 1D linear polynomial model of modeling base, along incorporating developed minimize matrix size techniques four digits base to encode residual efficiently and utilizing adaptive hierarchal quantization scheme.

2 Polynomial coding technique

Polynomial coding is one of the promising modern spatial techniques, characterized by simplicity of modeling concept and efficiency to compress images that are mathematically composed of two parts of deterministic and probabilistic bases of Taylor series [10]. In general, the technique impels the degree of approximation Taylor model (linear/nonlinear model) and block nature (2D/1D block base), here the linear approximation model of 1D adopted, to overcome the complexity of the 2D shape of the nonlinearity model, that leads to reduces the number of coefficients and computations [6].

1-D polynomial coding is a simple version of the polynomial coding techniques adopted by Al-Khafaji and George in 2021, to exploit interpixel (spatial) redundancy efficiently with a small number of coefficients and low computations [6], the techniques work to compress grey image of size $N \times N$ of high correlation embedded, using the following steps below.

Step 1: Use an original uncompressed grey square image I (8 bits/pixel) of bitmap image file (BMP) format, where the image size of 256×256 is adopted.

Step 2: Partition or segment I into 2D non-overlapping blocks of fixed nature each of size $n \times n$ I_{2D}. This step is essential need to represent each segmented block by a finite number of coefficients; here the fixed partitioning scheme is adopted for simplicity and popularity, despite its inefficiency compared to non-fixed schemes of homogeneity nature of quadtree, horizontal-vertical and triangular schemes. The number of partitioned (segmented) blocks (regions) equals $(N/n)^2$, here the restricted block size is equal to 4×4, hence the number of squared regions equals $64 \times 64 = 4096$ segments.

Step 3: Convert each segmented 2D block of partitioned image I2D, into a simple representation of 1D vector scheme I_{1D}, so the 2D block of $n \times n$ size (i.e., 4×4) is represented by 1×16 where I_{1D} is of 4096 16 blocks.

Step 4: Apply polynomial coding techniques of the 1D base, where the deterministic part that implies the estimated coefficients computed according to [6]:

$$a_{01D} = \frac{1}{n^2} \sum_{i=0}^{n^2-1} I_{1D}(i) \tag{2.1}$$

$$a_{11D} = \frac{\sum_{i=0}^{n^2-1} I_{1D}(i)(i - x_c)}{\sum_{i=0}^{n^2-1} (i - x_c)^2} \tag{2.2}$$

$$x_c = \frac{n^2 - 1}{2} \tag{2.3}$$

where a_{01D} coefficient represents the mean value of each one-dimension block (segment) of size $1 \times n^2$ (i.e., 1×16), a_{11D} corresponds to the ratio of pixel summation multiplied by the distance from the center (x_c) to the squared distance from the center (x_c) and n^2 corresponds to the block size (i.e., 16).

Obviously, by utilizing this 1D scheme, each block can easily be modelled by two coefficients (a_{01D}, a_{11D}), compared to a 2D scheme of three coefficients (a_0, a_1, a_2), that implicitly leads to improvements in compression performance (one coefficient discarded for each block) with reducing the computation necessary.

Step 5: Encode/decode the computed coefficients from the step above either lossily or losslessly, where the choice between them depends on the problem being solved.

Step 6: Create the predicted image $\tilde{I}_{1D}$ using the encoded coefficients, such as:

$$\tilde{I}_{1D} = a_{01D} + a_{11D}(i - x_c) \tag{2.4}$$

$\tilde{I}_{1D}$ is the predicted or modelled image, which is an approximation of the original image that created compactly using two coefficients (a_{01D}, a_{11D}), where the compression takes place, namely two coefficients for each block of size 1×16 adequate to create an approximation image.

Step 7: Find the probabilistic part that corresponds to the second part of the modelling base, which equals to the residual or (prediction error) that represents the difference between the original image I1D and the modelled one (predicted) $\tilde{I}_{1D}$

$$e = I_{1D} - \tilde{I}_{1D} \tag{2.5}$$

where e is the residual image.

Step 8: Encode/decode e either approximated (lossily) or identically (losslessly).

Step 9: Reconstruct or rebuild either identical or approximated image by adding the predicted to the residual images, implicitly means adding the two parts of deterministic and probabilistic basses, in this thesis the lossy scheme of linear approximation base is adopted that discussed in details in chapter three.

3 Minimize Matrix-Size Algorithm

This technique was introduced by M. Siddeq in 2010 [11], that basically represents losslessly each four data values into a single floating one by utilizing of weighted values (key) which generated randomly between [0-1], according to equation bellow [11]:

$$M_{M4}(i,j) = \sum_{t=1}^{Ms} K(t) \times M_4(i, j+m) \tag{3.1}$$

Where: Ms here equals 4, K random keys, $m = 0, 4, 8, 12, ...$ row size, i, j =row and column, $M4$ corresponds the image with redundancy, M_{M4} is the compressed minimized matrix of four data base which corresponds to the sum of products of weights and original data values. After that find the probability (occurrence) of the original data that referred as limited data (LD) which is unique data that essentially used for reconstruct the original values identically using the sequential search algorithm (SSA) [11].

4 Haar DWT

The simple Forward Haar basis Wavelet (FHWT) of 1D case is adopted in this thesis, as used in [13]:

If Z is even:

$$\begin{cases} L_i = \frac{I_{2i}+I_{2i+1}}{\sqrt{2}}, & i = 0...\left(\frac{Z}{2}\right) - 1 \\ H_i = \frac{I_{2i}-I_{2i+1}}{\sqrt{2}}, & i = 0...\left(\frac{Z}{2}\right) - 1 \end{cases} \tag{4.1}$$

If Z is odd

$$\begin{cases} L_i = \frac{I_{2i}+I_{2i+1}}{\sqrt{2}}, & i = 0...\left(\frac{Z-1}{2}\right) - 1 \\ H_i = \frac{I_{2i}-I_{2i+1}}{\sqrt{2}}, & i = 0...\left(\frac{Z-1}{2}\right) - 1 \\ L_{(Z+1)/2} = \sqrt{2} I(Z-1) & \\ H_{(Z+1)/2} = 0 & \end{cases} \tag{4.2}$$

Where Z is data size of input sequence (I_i), $i = 0...Z-1$, L_i and H_i corresponds to low pass filter and high pass filter respectively.

Lastly, each of techniques above has advantages and disadvantages, for SC the main advantage is the simplicity and the disadvantages the performance results directly affected by the image details (characteristics), unstandardized model, and estimation coefficients techniques, while for TC the main advantage is the highly packing, and the disadvantages related to the transformation utilized, shape and size of the block [5].

5 Related work

- Al-Khafaji and George [6], introduced a 1D polynomial scheme of the lossy grayscale base, that reduces the number of coefficients of a deterministic part to only two coefficients (a_{01D}, a_{11D}), along utilizing the non-uniform quantization lossily for the residual probabilistic part. The performance analysis used six natural and medical standard gray images of sizes (256×256) pixels, with block sizes of 4×4, and different quantization steps (Q) of values between 4 to 32 the compression ratio between 2 to 8 or exceed it on average according to image complexity with PSNR range between fifty or more to 38dB, also the test results implies a comparison with JPEG and traditional 2-D polynomial coding, that indicates higher quality compared to the other approaches, with nearly converge to JPEG compression performances and higher than the 2-D traditional approaches. These techniques suffer from a large residual size due to fewer coefficients utilization.

- While M. Siddeq [11], introduced the first methodology scheme of minimizes matrix size algorithm (MMSA) of four floating-point keys (weight) generated randomly with a limited range between 0 to 1, that implicitly reduces each four values to a single float one, which used to compress the grayscale image using the DWT and JPEG. The experiments results on two natural standard gray images square/non-square sizes, Lena (500×500) pixels, and the Animal (960×720) pixels, using weight values of $\{0.950, 0.231, 0.606, 0.485\}$, with quantization values between [1-20], indicated the better compression performance compared to the standards JPEG and JP2 where compression ratio attained between 3.77 and 10.25 for the first image, and between 3.53 to 12.19 for the second image, but with low-quality values compared to the standards between that 32.25dB and 30.69dB for Lena image, and 31.74 to 29.84dB for the animal image. Lastly, this technique suffers from a long encoding time due to the utilization of a sequential search algorithm.

- Samara S. Al-Hadithy et al. [4] introduce MMSA with C621 to compress residual of 1-D polynomial coding model testing on same images in Ghadah and Loay [6] high performance of CR between (9 to 11) with and PSNR values between (39-40) dB compared to Ghadah and loay [6].

6 Proposed system

6.1 Data sets

Four images used in this research are two data sets of natural color images that commonly used in all aspects of life and broadcast in the media, two images from USC-SIPI image (Miscellaneous) data set (Lena, Splash), these images are database at university of southern California-signal and image processing institute (USC-SIPI) is a collection of digitized images. It is largely maintained to facilitate image processing, image analysis, and machine vision research [12] and other from Kodak data set (Color Kid, Parrot) is widely used as a reference dataset in the domains of image compression and color image processing [13].

6.2 System layout of YC_bC_r base

Proposed system is applied on color transformation base of YC_bC_r bands, since the band Y contains the majority of the basic information in the image.

6.2.1 Pre-processing Steps

A simply pre-processing step is used in the preparation for compression processing that implies resizing the images into square size if necessary, separation and converting of the color image bands.

6.2.2 Source color band

Previously, it had been mentioned that the source color band (Y) I_{Sr} compressed based on spatial adaptive 1-D polynomial coding along the new C421 techniques, the subsections bellow divided into two parts, addressing the performance of polynomial coefficients and residual respectively.

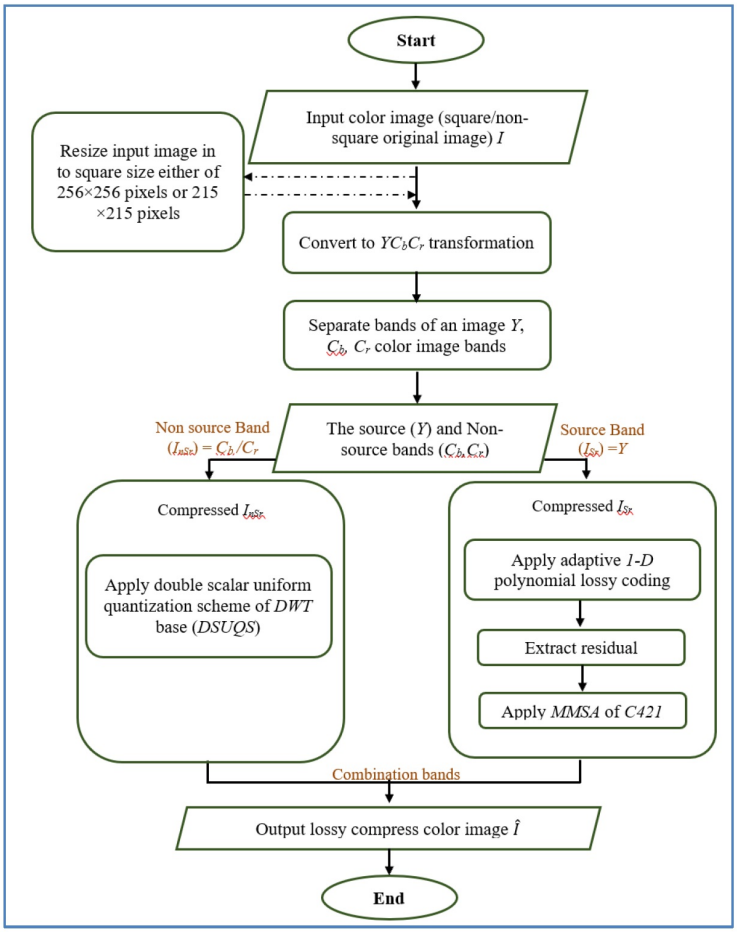

Figure 1: Flowchart of the proposed color compression system of YC_bC_r base.

Selected the source color band (Y) compressed differently from non-source color (C_b, C_r) bands, here for source band a new mixture lossy technique was presented of adaptive 1-D linear polynomial coding of spatial modeling base along improved MMSA technique of four values binary search scheme.

The source color image I_{Sr} is compressed by exploiting adaptive 1-D polynomial coding and C421 proposed system, steps (1-3) discuss in details the encoder process that implies five (DDPCM for $a_{01D}Q$, Zero/Non-Zero for $a_{11D}Q$, LD, $MM_{421Level2}$) encoded information (parameters), where implementation steps explained below.

Step1: Use source color band I_{Sr} of size $N \times N$.

Step2: Adaptive 1-D polynomial coding step: Apply spatial modeling technique to compress source band adaptively using sub steps bellow, also illustrated clearly in figure 2.

1. Partition I_{Sr} into non-overlapping fixed block size of 2D nature of size $n \times n$ (4×4) block size (region), then convert each region (segment) into 1D block nature of size $1 \times n^2 (1 \times 16)$.
2. Estimate the linear polynomial coefficients of 1-D base (a_{01D}, a_{11D}) according to equations (1-3).
3. Quantize polynomial coefficients lossily using traditional scalar uniform quantizer, where quantization step here equal to 1.

$$a_{01D}Q = Round(a_{01D}) \tag{6.1}$$

$$a_{11D}Q = Round(a_{11D}) \tag{6.2}$$

where (a_{01D}, a_{11D}) coefficients in 1-D polynomial coding technique, $(a_{01D}Q, a_{11D}Q)$ quantized coefficients.

4. Encode the quantized coefficients above adaptively to reduce the required bytes that consequently help for achieving better compression ratio, the following sub steps (i-iii) are obviously illustrated the adopted techniques in details:

i For quantized coefficients $a_{01D}Q$ that encoded using Double DPCM (DDPCM) to achieve higher decorrelation according to equations bellow, where the values of significant correlation, figure 1 shows the DDPCM clearly.

$$a_{01D}Q_{DPCM1} = a_{01D}Q(i) - a_{01D}Q(i+1) \tag{6.3}$$

$$a_{01D}Q_{DPCM2} = a_{01D}Q_{DPCM1}(i) - a_{01D}Q_{DPCM1}(i+1) \tag{6.4}$$

where $a_{01D}Q$ quantized coefficient a_{01D}, $a_{01D}Q_{DPCM1}$ quantized coefficient a_{01D} after first DPCM, $a_0Q_{1DDPCM2}$ quantized coefficient a_{01D} after second DPCM.

ii For quantized coefficients $a_{11D}Q$ that encoded using separating and counting techniques to efficiently improve the encoding process, to utilize the large zeros outcome of quantized a_{11D} values, figure 3 illustrate example explain this technique, which removes zeros and keeps track of the number of zeros between values. The outputs of this procedure are mapped into two vectors: one, counts the number of zeros and the other, counts the locations of non-zero values which should be easier and more efficient to compress, where this technique implicitly required a control parameter called range size (RS) that corresponds to the limited (specified) length of zeros.

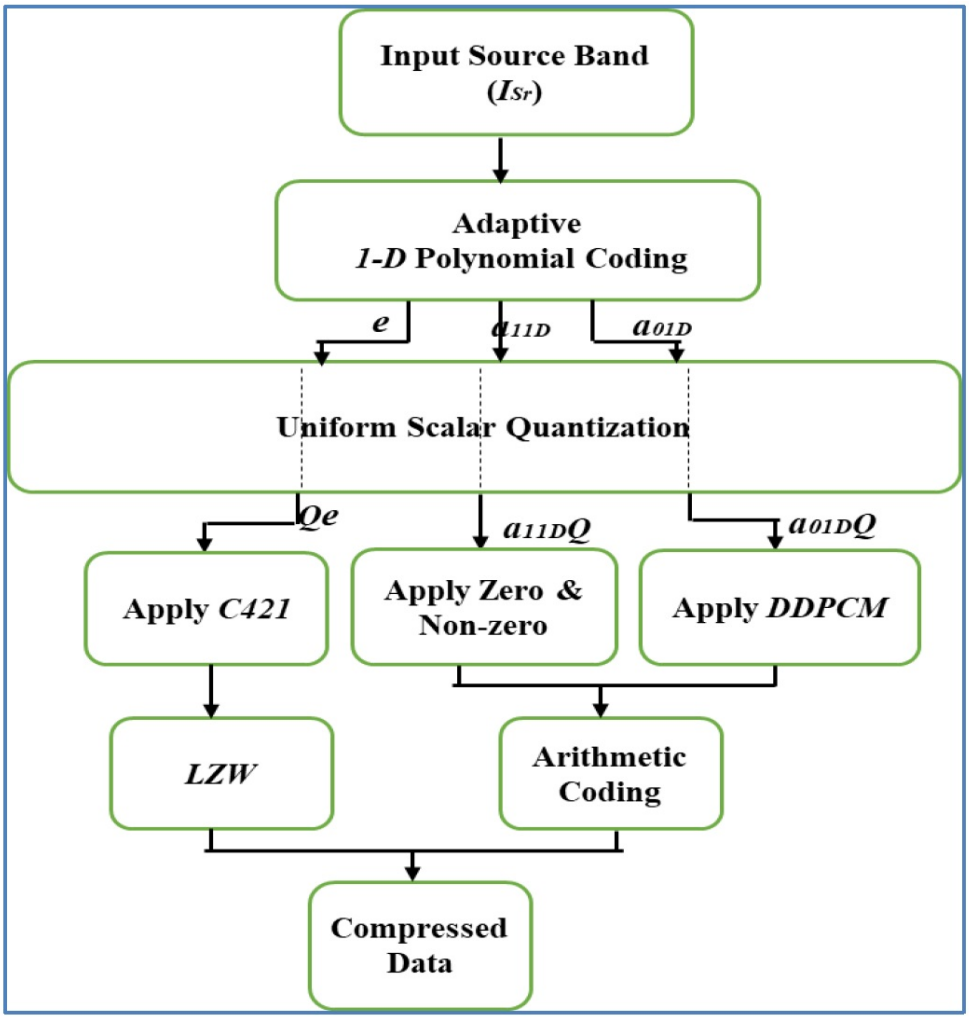

Figure 2: Adaptive 1-D polynomial coding technique

iii Encode the polynomial coefficients of DDPCM, zero and non-zero for quantized $a_{01D}Q, a_{11D}Q$ respectively using the arithmetic coding techniques.

5. Find residual image of probabilistic part using the created predicted image see equations (2.4)-(2.5), then quantize it uniformly using equation bellow.

$$Qe = Round\left(\frac{e}{Q}\right) \tag{6.5}$$

Where e refers to residual, Q is quantization step, Qe quantized residual.

Step3: C421 step for quantized residual image: Use new MMSA that utilized two-layer key generation scheme of integer base instead of one layer random floating point key, that explained clearly using sub steps bellow:

1. Create four integer keys (values) of two layers scheme, where

$$Key1_{Level1} = Start\ value_{Level1} \tag{6.6}$$

$$Max_{Level1} = Max(I_{Sr}) \times 2 \tag{6.7}$$

$$Key2_{Level1} = Key1_{Level1} + Max_{Level1} \tag{6.8}$$

$$Key1_{Level2} = Start\ value_{Level2} \tag{6.9}$$

$$Max_{Level2} = Max(Level1) \times 2 \tag{6.10}$$

$$Key2_{Level2} = Key1_{Level2} + Max_{Level2} \tag{6.11}$$

Start value is a selected integer number for two adopted levels, Max_{level1} is the maximum value of input I_{Sr}, Max_{level2} is the maximum value of results values in Level1.

2. Convert Qe into a vector then create the MMSA hieratically of C421 base using equations bellow:

$$Level_{1E1}(i) = \sum_{st=inc}^{QeS-2} Key1_{Level1} \times Qe(st) + Key2_{Level1} \times Qe(st+1) \tag{6.12}$$

$$Level_{1E2}(i) = \sum_{kt=2+st}^{QeS} Key1_{Level1} \times Qe(kt) + Key2_{Level1} \times Qe(kt+1) \tag{6.13}$$

where QeS corresponds to size of quantized residual, $Key1_{level1}$, $Key2_{level1}$ represents the two generated keys, $inc = 0, 4, 8, ...$ $Level_{1E1}$, $Level_{1E2}$ first level compression of four values Level1 first level of MMSA

$$MM_{421Level2}(i) = \sum_{st2=inc2}^{level1size} Key1_{Level2} \times Level1_{E1}(i) + Key2_{Level2} \times Level1_{E2}(i+1) \tag{6.14}$$

where $Key1_{level2}$, $key2_{level2}$ corresponds to the generated keys in level two, $Level_{1E1,E2}$ first level compression of four values based , $inc2 = 0, 2, 4, 6, 8, ...$, $MM_{421Level2}$ is the compressed data second level of two values based. The technique simply starts from the quantized residual image (Qe), representing the root after converted into a vector form, which corresponds to layer 0, then implementing the discussed C421 method. Here the $MM_{421Level2}$ corresponds to the second layer (layer 1); in order to construct the second layer, the coefficients from the previous layer are regarded as an input using the different keys values in the second current layer.

3. Preparing a limited Data table (LD) that consists of $MM_{421Level2}$ without redundancy, namely without repetition, and is ordered in ascending to be it's used in decoding stage to prepare it for binary search algorithm.
4. Encode the C421 parameters (LD,$MM_{421Level2}$) using dictionary-based techniques of LZW base to acquire more compression performance. Figure 4 shows example of C421, where the input is values of Qe, to generate the keys depended on equations (6.2)-(6.7) and (i.e maximum value in I_{Sr} is 280).

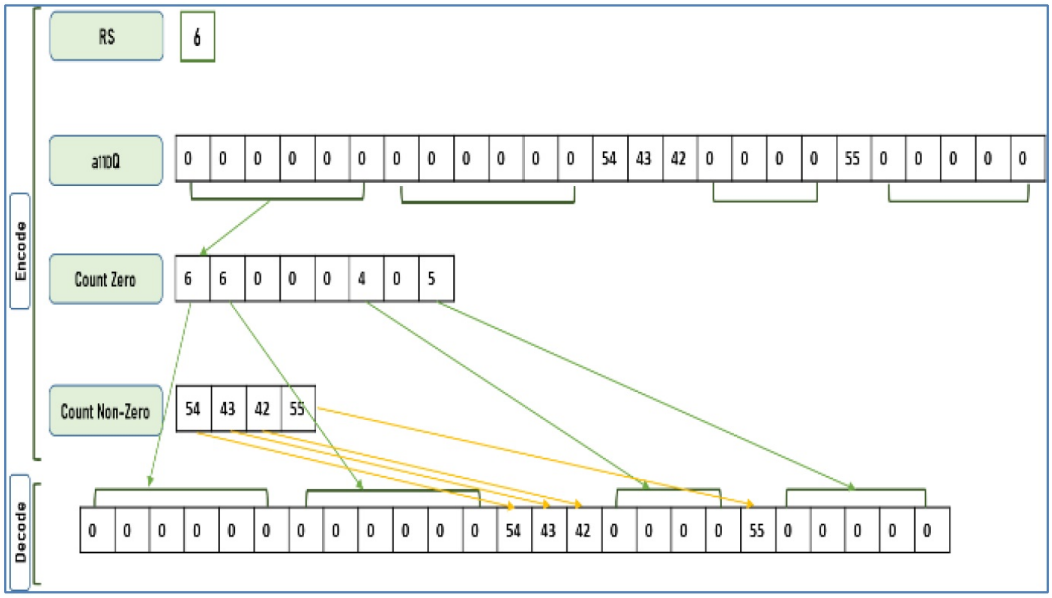

Figure 3: Example of Zeros and Non-zero splitting algorithm for encode $a_{11D}Q$ coefficient.

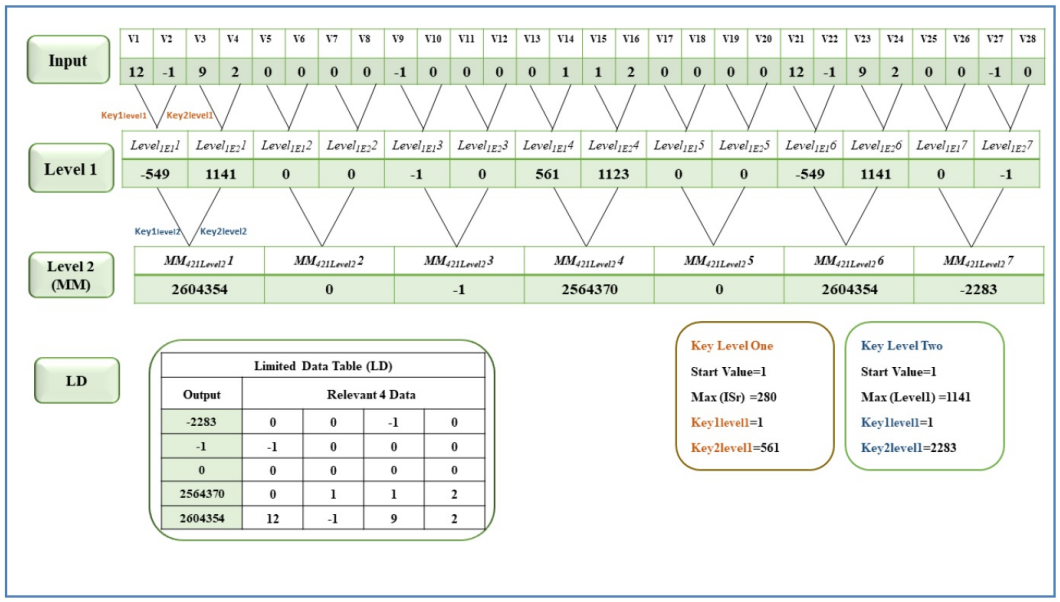

Figure 4: Example of C421 encoding techniques.

6.2.3 Non-Source Color Band Compression Steps

In the proposed YC_bC_r compression scheme the non-source color bands compressed differently using lossy double scalar uniform quantization scheme that exploited the DWT of hierarchal base and traditional one with the suggested C421 techniques to compress the residual efficiently.

Here, an adaptive method called double scalar uniform quantization scheme (DSUQS) is presented with high performance in terms of quality and compression ratio.

Step 1: Use one of the non-source color image band I_{nSr} of size $N \times N$.

Step 2: Apply Haar DWT of three levels hierarchal decomposition scheme [1].

Step 3: Utilize adaptive hierarchal scalar uniform quantization process as in [8], that implies four control parameters that vary according to the quadrants nature, which implies Quantization step (Qw), alpha (α), Beta (β) and Gamma (λ), where the quantization step for each levels computed (the quantization step is increase with the higher

level of wavelet).

$$Qstep = \begin{cases} Qw\alpha^{w-1} & for\ LH\ in\ w^{th}\ level \\ Qw\beta\alpha^{w-1} & for\ HL\ in\ w^{th}\ level \\ Qw\beta\lambda\alpha^{w-1} & for\ HH\ in\ w^{th}\ level \end{cases} \tag{6.15}$$

where $(Qw, \alpha, \beta, \lambda)$ the parameters of quantization step, LH, HL, HH refers to sub bands of image in Haar DWT.

Step 4: Encode the quantized details quadrants values ($LH_1, HL_1, HH_1, LH_2, HL_2, HH_2, LH_3, HL_3$ and HH_3) and the approximation quadrant ($LL3$), losslessly using LZW techniques.

Step 5: Create the lossy image using equations (6.13) hierarchally $\tilde{I}_{nSr}$, then find the residual between the created image and the original non-source band

$$e_{nSr} = I_{nSr} - \tilde{I}_{nSr} \tag{6.16}$$

where $\tilde{I}_{nSr}$ is the predicted non source band, I_{nSr} is the original non source band, e_{nSr} refer to residual non source band.

Step 6: Quantize the residual image e_{nSr} uniformly more details in [4].

Step 7: Encode the quantized residual image using C421 techniques, where the compressed information (LD, $MM_{421Level2}$) coded using LZW.

6.2.4 Compression System Decoder of YC_bC_r base

The decoder is the builder of the compressed information to reconstruct image back after the redundancy(s) eliminating. In other words, the decoder utilized the compressed (coded) information to recover the compressed image, here the decoder implies three steps, the first one for reconstruct the source band, the second is reconstructed of non-source bands and finally reconstruct the compressed image back, figure 5 and the steps bellow show retrieve the compressed image.

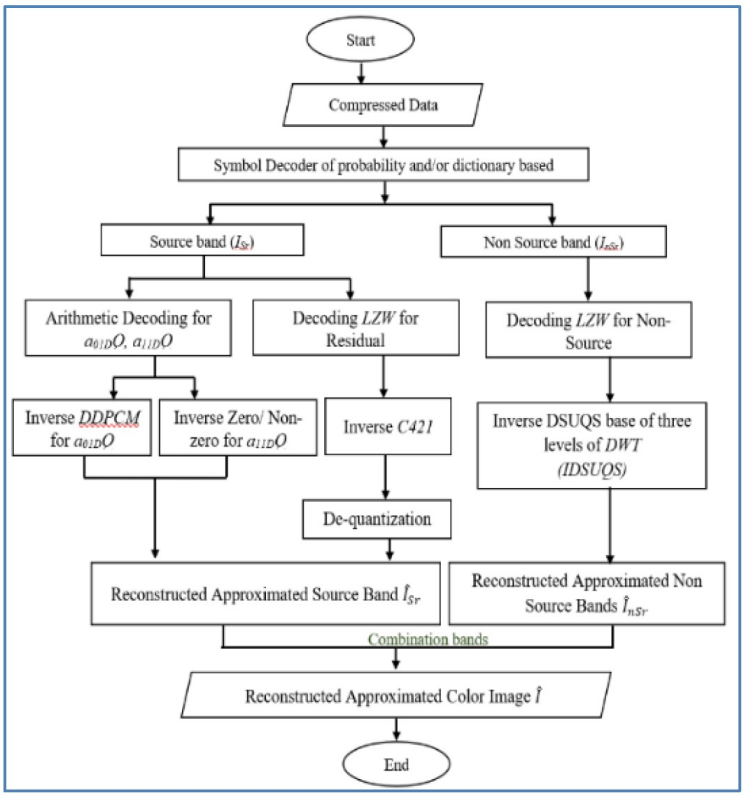

Figure 5: Flowchart of the YC_bC_r decoder system

Step1: Source band reconstruction Step: For source band the compressed coded information represented by 5 parameters for linear 1-D polynomial and C421 techniques (DDPCM of a_{01D}, Zero/Non-zero of a_{11D}, LD, $MM_{421Level2}$) that decoded firstly, then the following sub steps used, such as:

1. Reconstruct a_{01D} values using inverse DDPCM (IDDPCM), as in equations (4.1)-(4.2).
2. Re-expanding the a_{11D} coefficients from Zero/Non-zero coefficients.
3. Dequentize the residual image using the decoded C421 information, here the binary search adopted to overcome the time consuming of SSA base techniques that adopted previously, that clearly shown illustrative example in figure 6.

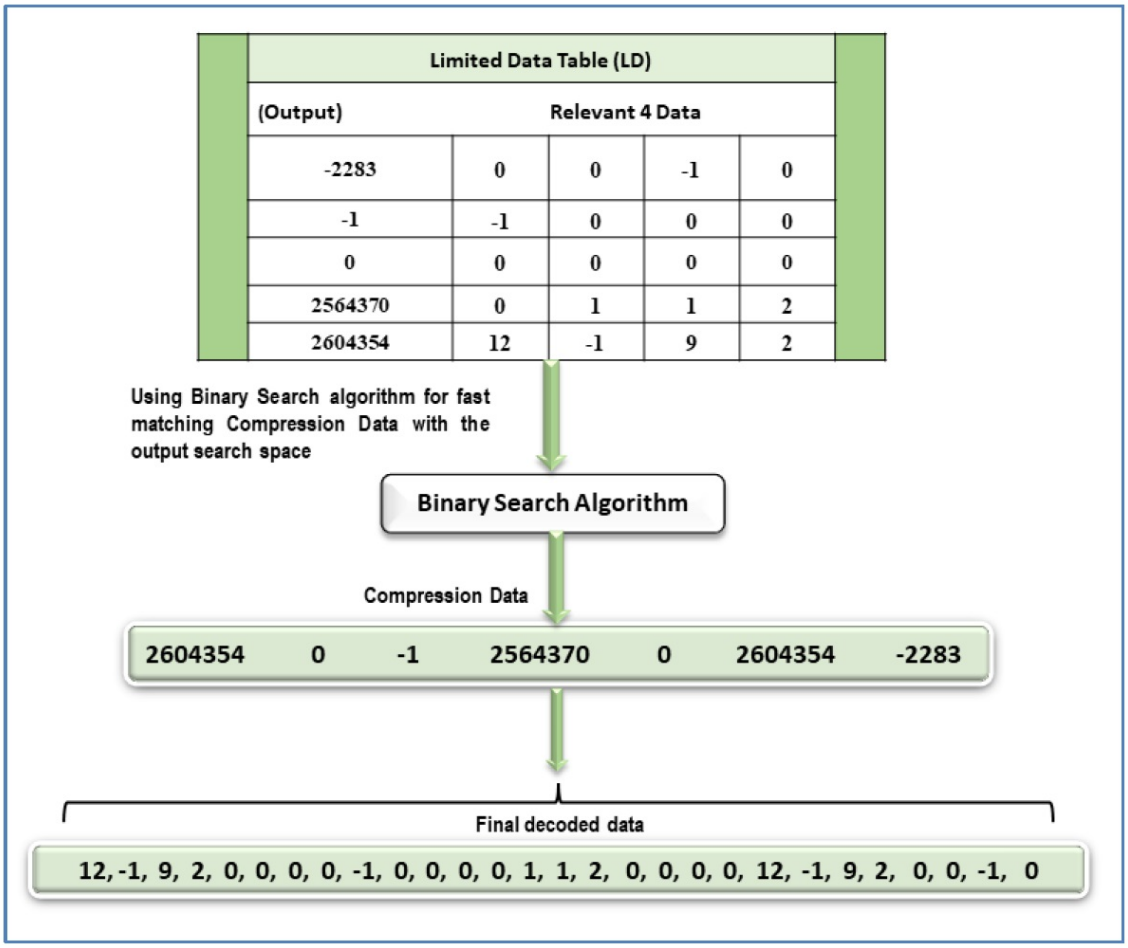

Figure 6: Example of C421 decoding technique.

In LD table consists of five columns, which consists of the first column, the compressed data without repetition, and the rest of the columns are the basic data on which the MMSA process was carried out, here to complete the example in figure 6 where the 1st row (0,0,-1,0) that yields -2283, also for the 2nd row (-1,0,0,0), 3rd row (0,0,0,0) yields (0) and (0,1,1,2) yields (256370), then binary search algorithm is also used in the process of matching the output of the table with the compressed data ($MM_{421Level2}$), and when the matching is completed, the four data formed from it are downloaded. Namely, the 1st number (2604354) comes from four digits (12,-1,9,2) then (0) also comes from (0,0,0,0) and so on, so all the number reconstructed eventually.

Reconstruct the approximated compressed source band using the dequantized residual image that adds to the predicted one that created using the decoded polynomial coefficients, use equations (6.4)-(6.5) respectively.

Step 2: Non-Source band reconstruction Step:

For the non-source band the compressed coded information is represented by 12 parameters (9 for details sub-bands, 1 approximation sub band and 2 for C421), for DSUQS and C421 techniques that decoded firstly. Then the following sub-steps are utilized:

1. Dequentize the residual image using equation (6.2) using the decoded C421 information.
2. Apply the inverse of Haar DWT of hierarchal dequantization scheme.
3. Reconstruct the approximated compressed non source band by adding the dequantized residual image to the inverse of DWT, according to the equation bellow.

$$\hat{I}_{nSr} = De_{nSr} + \tilde{I}_{nSr} \tag{6.17}$$

where De_{nSr} dequantize residual of non-source band, $\tilde{I}_{nSr}$ is reconstricted of non-source band and $\hat{I}_{nSr}$ is approximated non-source band.

Step 3: Reconstruct the color image by adding all color band together, namely the source and non-source color bands.

7 Results

The first experiments related to the estimated $a_{01D}Q$ coefficients encoding of 1-D polynomial coding (equation (2.1)) lossily, where the adopted techniques imply rounding, followed by DDPCM (equation (6.2)) and coded using arithmetic coding techniques. Tables 1 which compares the size in bytes for two data sets tested images using the traditional techniques solely of probability base that implies Huffman/Arithmetic coding and the adopted techniques, where the size in bytes for a_{01D} coefficients without coding equals to 4096 bytes (i.e., 1×4096 bytes) or 16384 for (256×256) or (512×512) pixels respectively, with block size of 4×4 (1×16).

Table 1: A comparison performance of $a_{01D}Q$ coefficients encoding between the proposed technique of DDPCM to the traditional techniques of probability based for natural images of (Miscellaneous, Kodak), in YC_bC_r compression system.

Tested Images	**Size in bytes required using Huffman coding techniques**	**Size in bytes required using Arithmetic coding techniques**	**Size in bytes required using proposed (DDPCM)**
Lena	14220	14165	6523
Splash	14378	14317	5742
Color Kid	14942	14873	6338
Parrot	14488	14444	5974

The second experiments related to the estimated $a_{11D}Q$ coefficients encoding of 1-D polynomial coding (equation (2.2)) lossily, where the adopted techniques imply rounding, followed by splitting zero/non-zero with different RS values that coded using arithmetic coding techniques. Tables 2 which compares the size in bytes for two data sets tested images using the traditional techniques solely of probability base that implies Huffman/Arithmetic coding and the adopted techniques, where the size in bytes for $a_{11D}Q$ coefficients without coding equals to 4096 bytes (i.e., 1×4096 bytes) or 16384 for (256×256) or (512×512) pixels respectively, with block size of 4×4 (1×16) along utilizing RS values equals to 5,10, and 20.

Table 2: A comparison performance of $a_{11D}Q$ coefficients encoding between the proposed technique of Split Zero and Non-zero to the traditional techniques of probability based for natural images of (Miscellaneous, Kodak), in YC_bC_r system

Tested Images	**Size in bytes required using Huffman coding techniques**	**Size in bytes required using Arithmetic coding techniques**	**Size in bytes required using proposed (DDPCM)**		
			RS=5	RS=10	RS=20
Lena	2566	1616	1673	1592	1529
Splash	2422	1222	1244	1160	1084
Color Kid	2532	1494	1505	1446	1381
Parrot	2432	1194	1135	1066	996

Results in tables above clearly show that the traditional techniques of probability base (Huffman/Arithmetic) converge to each other with small differences, with superior performance of the proposed techniques due to exploiting embedded redundancy efficiently. but unfortunately, when utilizing larger RS values (i.e., 30,40,50) the difficulty of finding such length runs of zeros that leads to very slightly differences in bytes or sometimes to more size in bytes.

7.0.1 Total size in Bytes of Polynomial Coefficients ($a_{01D}Q, a_{11D}Q$)

The extended continuity of the two experiments above that attempted to measure the size of deterministic part, which involved the encoding compressed ($CS_{Deterministic}$) sized in information of $a_{01D}Q, a_{11D}Q$ in bytes that which are coded lossily, so:

$$CS_{Deterministic} = [Sizebytes\ for\ a_{01D}Q\ coefficants + Sizebytes\ for\ a_{11D}Q\ coefficants] \tag{7.1}$$

Table 3: Size of deterministic part after coding in proposed system in bytes (Miscellaneous, Kodak), in YC_bC_r system.

Tested Images	$a_{01D}Q$	$a_{11D}Q$	$CS_{Deterministic}$
Lena	6523	1529	8052
Splash	5742	1084	6826
Color Kid	6338	1381	7719
Parrot	5974	996	6970

C421 Residual Encoding/Decoding Results. The third experiment related to the residual image or probabilistic part, which is the difference between original image and the predicted one (equation (2.5)), Table (4) shows the results of quantizing and coding the residual images of C421 lossily with start $value_{Level1}$ equal to 1, see equation (6.4), where the range of attained uniform quantization step value (Q), is limited between 1 to 12. The PSNR between original and residual image (Res) and the dequantized image (ResD) is also computed, along the encoding/decoding time in (sec) with total size in bytes, such as:

$$CS_{Probablistic} = [Sizebytes\ for\ LD + Sizebytes\ for\ MM_{421level2}] \tag{7.2}$$

Table 4: C421 performance in terms of $CS_{Probablistic}$, PSNR and encode/decode time of residual images in YC_bC_r system for (Miscellaneous, Kodak) dataset.

Tested Images	Q	$CS_{Probablistic}$	**PSNR (Res,ResD)**	**Encode/ Decode Time (sec)**
Lena	1	54410	62.853	101.681
	2	33251	45.273	34.019
	5	11523	39.134	4.323
	7	10146	36.294	3.577
	10	8873	33.518	1.539
	12	8142	32.329	1.252
Splash	1	36915	64.599	41.977
	2	15634	45.252	8.921
	5	10518	39.399	1.707
	7	9660	36.801	1.177
	10	8368	34.548	0.961
	12	7408	33.704	0.913
Color Kid	1	54062	63.543	84.449
	2	36471	45.311	42.638
	5	13446	39.708	6.306
	7	10822	37.154	3.331
	10	9472	34.476	1.922
	12	8577	33.202	1.486
Parrot	1	45859	65.042	50.510
	2	29684	45.219	21.802
	5	11694	39.477	3.241
	7	9805	37.018	2.145
	10	8388	34.609	1.579
	12	7526	33.549	1.343

As expected, the behavior of the residual quantizing/dequantizing and C421 coding directly affected by Q values, in terms of residual size in bytes, PSNR and encoding/decoding time, where the size of residual varies with Q, and the residual size with the PSNR and time inversely related to the Q values.

The second part of the third experiment relates to comparison the adopted C421 with the other MMSA of hexa-coding base that referred as C621 which adopted by S.S. AL-Hadithy [4] of floating point keys, table 5 shows the results of quantizing and coding the residual images of C621 lossily, using Q equals 7 for the two tested datasets.

Table 5: Comparison between C421 and C621 for natural image of (Miscellaneous and Kodak) dataset in YC_bC_r system

Tested Images	$CS_{Probablistic}$	**Time**
Lena	20052	9.607
Splash	11154	19.116
Color Kid	23725	11.794
Parrot	16764	16764

Obviously, the C621 and C421 aimed to encode residual efficiently, but the results showed that measures in terms of $CS_{Probablistic}$, time reduced to nearly converged to half with faster encoding/decoding time with same quality respectively for C421 compared to C621.

The Fourth experiments concerned with compressing the non-source color bands of YC_bC_r base (C_b, C_r), that encoded using the proposed system of DSUQS, tables 6 shows the performances in terms of PSNR and CR, with the quantization steps utilized were selected to be Q_{nSr} between 2 to 10, Qw to be 20, along quantization coefficients of best experimental test with $\alpha = 3.9, \beta = 1.2, \lambda = 2.1$ of best.

Table 6: DSUQS performance for non-source color band in (Miscellaneous) dataset for YC_bC_r system.

Tested Images	**Non-Source**	$Q_{nSr} = 2$		$Q_{nSr} = 5$		$Q_{nSr} = 7$		$Q_{nSr} = 10$	
		CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR
Lena	$I_{nSr}1$	19.074	47.327	24.136	40.703	26.306	38.661	29.394	36.797
	$I_{nSr}2$	17.196	47.168	21.229	40.248	23.171	38.147	25.788	36.271
Splash	$I_{nSr}1$	21.664	46.921	29.775	40.255	36.823	38.688	45.653	37.474
	$I_{nSr}2$	18.070	46.885	20.734	39.358	24.054	37.213	29.391	35.530
Color Kid	$I_{nSr}1$	25.723	47.296	39.736	41.578	46.620	40.060	56.411	38.868
	$I_{nSr}2$	20.605	47.517	28.139	41.265	30.895	39.236	34.253	37.429
Parrot	$I_{nSr}1$	22.999	47.066	32.600	40.661	38.191	38.836	46.298	37.407
	$I_{nSr}2$	22.678	47.213	32.439	41.058	36.890	39.254	43.315	37.736

The results showed highly performers in YC_bC_r base in terms of CR and PSNR values using DSUQS of mixing techniques of DWT and C421. Also the Q_{nSr} of compromise 10 value adopted for comparison the standard JPEG.

The combination of compressed color bands information to reconstruct the approximated color compressed image of YC_bC_r base, where complementary computation required in terms of CR, and PSNR, here the measures adopted by the following formulations:

$$CompAllBands = (Comp_{ISr} + Comp_{InSr1} + Comp_{InSr2}) \tag{7.3}$$

$$CRColor = (OriginalBands/CompAllBands) \tag{7.4}$$

$$PSNRav = \left(\frac{PSNR_{ISr} + PSNR_{InSr1} + PSNR_{InSr2}}{3}\right) \tag{7.5}$$

Where $CompAllBands$ size in bytes for all compressed bands, $Comp_{ISr}$ is compressed source band in bytes, $Comp_{InSr1}$, $Comp_{InSr2}$ are compressed non source bands in bytes, $CRColor$ compression ratio for all color band in image, PSNRav average PSNR for all bands, $PSNR_{ISr}$ source band PSNR, $PSNR_{InSr1}$, $PSNR_InSr2$ are PSNR for non-source bands. Figure 7

8 Comparison of the Proposed System with JPEG

The last experiment compares the proposed compression system of YC_bC_r base against the commonly standard image compression techniques of JPEG XnView software, where the comparison implies the CR and PSNR measures

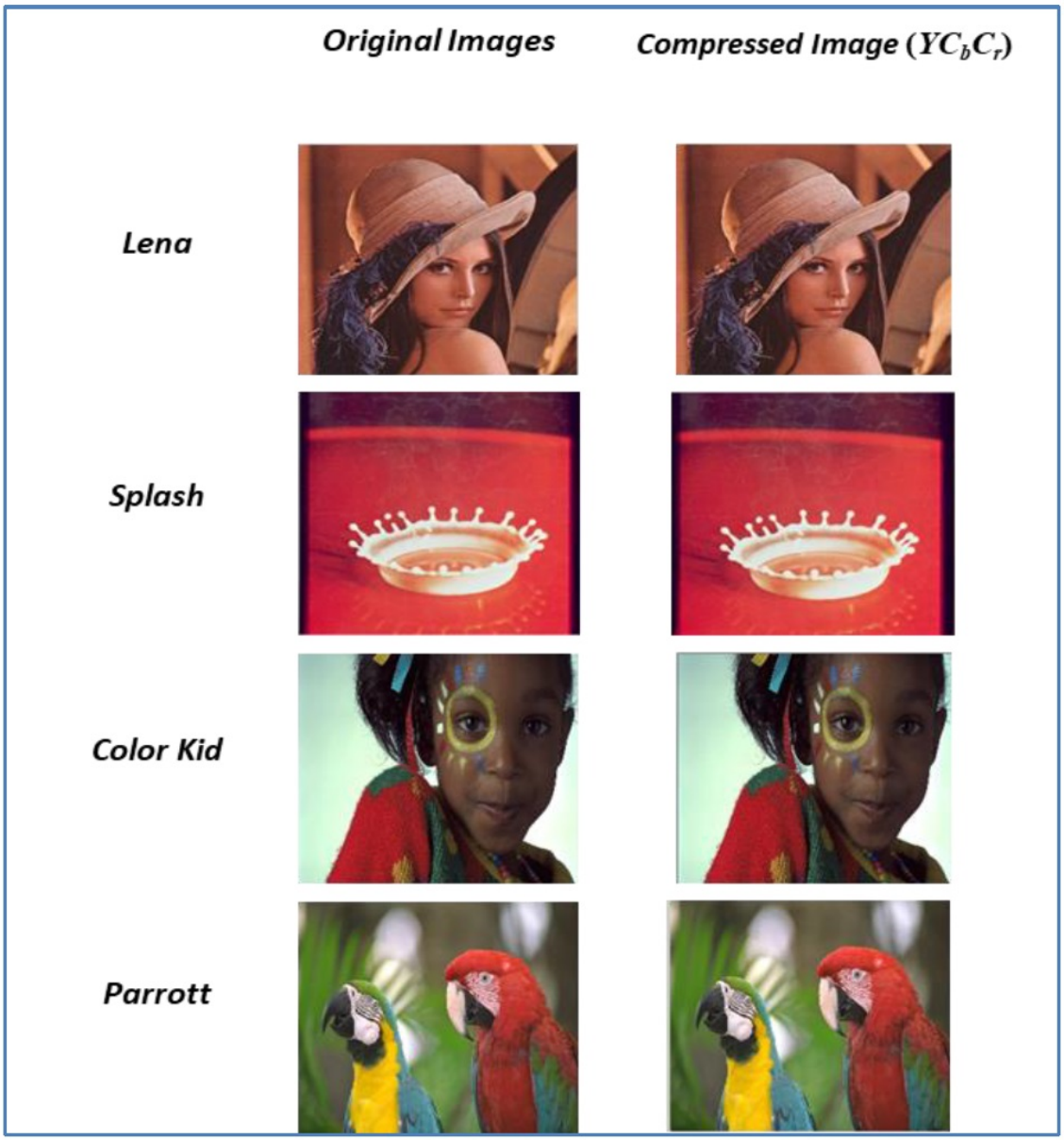

Figure 7: Original and compressed images (Miscellaneous and Kodak) in YC_bC_r compression systems.

as illustrated in tables 7,8 for the adopted two data sets using with blocksize=4, RS=20, Start $value_{Level1}$, Start $value_{Level2}$ equal 1, quantization $Q = 7, Q_{nSr} = 10, Qw = 20$. Which clearly showed the higher performances of the proposed system compared to the JPEG. Where JPEG the latter performance in terms of CR between (14-22) and PSNR between (29-38).

Table 7: Comparison standard JPEG with YC_bC_r proposed compression system for (Miscellaneous) dataset.

Tested Images	**Proposed System** (YC_bC_r)		
	CompTotal	CRColor	PSNRav
Lena (786432)	37290	21.089	36.454
Splash (786432)	31159	25.239	36.602
Tested Images	**JPEG**		
	CompTotal	CRColor	PSNRav
Lena (786432)	46387.2	16.953	30.810
Splash (786432)	35737.6	22.005	32.297

Table 8: Comparison standard JPEG with YC_bC_r proposed compression system for (Kodak) dataset.

Tested Images	**Proposed System** (YC_bC_r)		
	CompTotal	CRColor	PSNRav
Color Kid (786432)	30850	25.492	37.817
Parrot (786432)	28498	27.596	37.387
Tested Images	**JPEG**		
	CompTotal	CRColor	PSNRav
Color Kid (786432)	45977.6	17.104	30.859
Parrot (786432)	36966.4	21.274	32.703

9 Conclusions

1. The use of color transformation of YC_bC_r base efficiently improves the performances of the source suggested compression system.
2. Use RGB without converting to color transformation and select the source band.
3. The exploitation of adopted coding techniques of DDPCM and split zero/non-zero for 1-D polynomial coefficients $(a_{01_D}Q, a_{11_D}Q)$ achieved an acceptable reduction in bytes required with better performance of YC_bC_r base about 1KB (1000 bytes) on average.
4. C421 of two levels integer keys efficiently compress the residual image with no complication and fast compression/decompression process (encode/decode), which is directly affected by Q value, along the superiority of YC_bC_r.
5. The combination of C421 with DSUQS of three-level Haar DWT hierarchal scheme to compress (code) the non-source band effectively work with control parameters $(\alpha, \beta, \lambda, Qw, Q_{nSr})$.

10 Future work

- Use variable block sizes (quadtree, horizontal/vertical) instead of fixed partitioning blocks, along a variable number of blocks according to block nature for 1-D polynomial coding, with the possibility to extend the model into a non-linear 1-D scheme.
- Use non-uniform quantization with MMSA of C421 and compare the results.
- Extend the system to medical and facial (personal) color images of square/non-square sizes, in addition to incorporating the segmentation process if necessary.
- For non-source band use other DWT filters (Daubechies, EZW, SPIHT), different hierarchal schemes, along the mixture of uniform quantize and thresholding techniques with comparing the results.
- Use other efficient encoder mixing systems to code the compress the filtered information after filtering out of redundancy(s), with exploiting other color transformation systems that are compatible with an image used (YUV, YP_bP_r) and other statistical measures for selecting the source band (cross-correlation).
- For MMSA uses a multilayer integer key generated technique.

References

[1] A. Abdullah, *Photo passport compression using hybrid techniques*, MSc. Thesis, College of Science, University of Baghdad, Baghdad, Iraq, 2021.

[2] A. Abdullah and G.K. AL-Khafaji, *A pixel based method for image compression*, Tikrit J. Pure Sci. **26** (2021), no. 1, 113–122.

[3] A.A. Abdulelah, S.H.A. Hamed, M. Rasheed, S. Shihab, T. Rashid and M.T.K. Alkhazraji, *The application of color image compression based on discrete wavelet transform*, J. Al-Qadisiyah Comput. Sci. Math. **13** (2021), no. 1, 18–25.

[4] S.S. AL-Hadithy, *Adaptive 1-D polynomial coding of C621 base for image compression*, Turkish J. Comput. Math. Educ. **12** (2021), no. 13, 5720–5731.

[5] G.K. Al-Khafaji, *Intra and inter frame compression for video streaming*, Ph.D. Thesis, Exeter University, UK, 2012.

[6] G.K. Al-Khafaji and L.E. George, *Grey-level image compression using 1-D polynomial and hybrid encoding techniques*, J. Eng. Sci. Technol. **16** (2021), no. 6, 4707–4728.

[7] I.M. Krishna, C. Narsimham and A.S.N. Chakravarthy, *An effective lossy color image compression using multi transforms*, i-manager's J. Image Process. **8** (2021), no. 1, 12–19.

[8] S.A. Mahmood, *Using discrete cosine transform to encoding approximation wavelet subband*, MSc. Thesis, Al-Nahrain University, Collage of Science, Baghdad, Iraq, 2008.

[9] A.J. Qasim, R. Din and F.Q.A. Alyousuf, *Review on techniques and file formats of image compression*, Bull Electr. Eng. Inf. **9** (2020), no. 2, 602–610.

[10] A.-T. Rasha, *Intra frame compression using adaptive polynomial coding*, MSc. Thesis, University of Baghdad, Collage of Science, Iraq, 2015.

[11] M. Siddeq, *JPEG and sequential search algorithm applied on low-frequency sub-band for image compression (JSS)*, J. Info. Comput. Sci. **5** (2010), no. 3, 163–172.

[12] https://sipi.usc.edu/database/database.php.

[13] http://r0k.us/graphics/kodak.