International Journal of Engineering

Journal Homepage: www.ije.ir

Adaptive Polynomial Coding of Multi-base Hybrid Compression

G. K. AL-Khafaji[a], M. H. Rasheed*[b], M. M. Siddeq[b], M. A. Rodrigues[c]

[a] *Department of Computer Science, College of Science, University of Baghdad, Iraq*
[b] *Computer Engineering Department, Technical College/Kirkuk, Northern Technical University, Iraq*
[c] *GMPR-Geometric Modelling and Pattern Recognition Research Group, Sheffield Hallam University, Sheffield, UK*

PAPER INFO

Paper history:
Received 04 August 2022
Received in revised form 11 October 2022
Accepted 15 October 25, 2022

Keywords:
Image Compression
Lossless/Lossy
Polynomial Coding
Iterative Based Technique

ABSTRACT

With increasing demand for the intensive use of images, especially linked to online applications as well as the massive, continuous revolution of mobile phone technology, the need has emerged for efficient, standard image compression techniques that ensure simplicity and speed. These must be compatible with user needs, but also meet the challenges of improving compression techniques. Polynomial coding is one such techniques still under development, based on a modelling concept of deterministic and probabilistic coding bases. This paper introduces a new mathematical iterative polynomial model to represent both coding bases. The model proposes an efficient hybrid way where coefficients are represented as lossless while residuals are presented as a lossy but with minimum loss, which ensures effective performance in terms of compression ratios and quality. Results show that while the technique has some limitations, the proposed system achieves equivalent compression ratios as the standard JPEG technique, but with superior quality for the same compression ratio.

***doi**: 10.5829/ije.2023.36.02b.05*

1. INTRODUCTION

Today the number of people that are active online exceeds 2.5 billion. The vast majority use instant messaging (e.g., Viber, WhatsApp) and social media (e.g., Facebook, Twitter, Instagram), which can change our lives, relations, and even political views. Since we digitally communicate through data streams, conveying events (news), broadcasting TV, cinema and other media in cheap and effortless ways has become a must. The basic elements of these electronic communications are text messages, audio, video and images, and these need to be compressed to save excessive byte consumption (storage) and overcome limited bandwidths.

Generally, image compression reduces the required bits to represent an image through efficient exploitation of redundancy in the image itself. Redundancy utilization can be purely statistical or combined with psycho-visual effects [1] implying lossy and lossless techniques. To remove redundancy from the data implies transform coding (TC) and spatial coding (SC) along with mixtures of both called hybrid coding (HC). The background information related to compression basics can be found in literature [2-5], also reviews of various image compression techniques are described in literature [6-10]. Each technique has its own characteristics in terms of performance which is normally optimized for compression ratios and/or preserving image quality.

Today, due to their high performance, the dominant standard image compression techniques are the joint photographic expert group (JPEG) and JPEG2000 (JP2). Both employ lossy approaches that effectively utilize the TC of discrete cosine transform (DCT) and discrete wavelet transform (DWT), respectively [11, 12]. However, the need for efficient compression techniques means that this field is not yet mature and still represents an attractive research area. Techniques that use SC may compete with these standards. Predictive coding (PC), also referred to as auto-regression (AR), or differential pulse code modulation (DPCM) are used by a large number of research projects characterized by their simplicity, but still faces a number of inherent problems

*Corresponding Author Institutional Email: Mohammed.rasheed@ntu.edu.iq (M. H. Rasheed)

Please cite this article as: G. K. AL-Khafaji, M. H. Rasheed, M. M. Siddeq, M. A. Rodrigues, Adaptive Polynomial Coding of Multi-base Hybrid Compression, ***International Journal of Engineering, Transactions B: Applications***, Vol. 36, No. 02, (2023), 236-252

that can be summarized as: the difficulty of choosing an appropriate model, where the model is composed of three elements termed by order (number of neighbours), structure (1D/2D), causality (causal/acausal), the way of estimating the coefficients (linear/nonlinear) and the seed values (initial condition).

Polynomial coding solves the above-mentioned problems related to predictive coding techniques using *Taylor series*, where the model and the estimation coefficients methods are determined either by linear or non-linear models. It solves the approximation base with no use of seed values which can be considered the most pressing problem. Currently, polynomial coding is utilized to compress both lossy or lossless images [13-18], but still suffers from large residuals (prediction errors) and large number of coefficients.

This paper introduces a novel adaptive technique for lossy polynomial base to efficiently represent the coefficients and residuals applied independently to each image plane demonstrated as grey images. In other words, the work implies investigation into an innovative approach to model the deterministic and stochastic polynomial parts effectively using fewer required number of bytes using an iteration base scheme of high precision techniques, where a mathematical model is generated based on subtraction and division for coefficients (a0, a1, a2) and residual, respectively ensures the effectiveness in compression ratios and quality. The rest of the paper is organized as follows: section 2 reviews related work, section 3 describes the proposed technique, section 4 delivers experimental results with discussion, while conclusions are presented in section 5.

2. RELARED WORK

Polynomial coding is one of the modern techniques that overcome the inherited problems of predictive coding which is characterized by simplicity and symmetry, but still suffering from large byte consumption. Here we concentrate on a linear lossy polynomial approach used to compress greyscale images efficiently. The works surveyed here can be classified into two major classes: the enhancement-based polynomial which aims to improve the standard techniques with an adaptation process, and a residual-based technique which is concentrated on utilizing various residual quantization methods, where the residual can be considered the largest and main problem related to polynomial coding.

The first type of enhancement-based polynomial approach includes Ghadah [14], utilizing variable block sizes ($n \times m$) using the quadtree scheme instead of a fixed partitioning process of ($n \times n$). Variable square block sizes are adopted after determining the minimum and maximum block sizes, with a homogeneity measure and quantization step of coefficients. Concerning residuals, results are promising for standard natural images compared to traditional polynomial coding of fixed block size (4×4). Using smaller blocks of variable sizes ($Min=2$ and $Max=16$) the same performance is obtained in terms of quality and compression ratios. Athraa [12], exploited the hierarchical scheme of interpolation base, where the multi-resolution principle was adopted for three layers. Through enlarging or shrinking of nearest neighbour interpolation technique, a quarter of the image is compressed instead of the full image (i.e., quarter size of coefficients and residuals). Results were shown to be adequate and improved almost four times on the traditional model. Rasha [7], adopted three improvement techniques to enhance the polynomial coding. First, a hierarchal scheme was used in which the polynomial coefficients of the first layer were utilized efficiently to construct the second layer polynomial coding. Second, a fixed predictor was used to remove the spatial redundancy before utilizing the polynomial coding, and lastly the residual reduction was achieved using the discrete wavelet transform (DWT). All these adaptations aimed to overcoming the polynomial problems of redundancy embedded within the image itself, the coefficients, and residuals. The results show high performance compared to traditional polynomial based techniques with at least two times improvement in compression ratios on average while preserving high image quality. Murooj [13], used various fixed predictor models of certain order with different structures (1D/2D) on a causality basis to remove the inherited spatial redundancy embedded within the image, before using the polynomial coding to lossy compress a natural standard image. The approach also exploited the selective predictor model where each block utilized different predictors according to residuals. The results indicated improvements of four-fold increase in compression ratios while preserving image quality.

The second type of enhancement-based polynomial approach relates to the quantization process of the residual image, where block size is of 4×4 and the quantization coefficients is of scalar uniform base. These include: Ghadah [14], which quantized the residual image using block truncation coding (BTC) of binary representation, namely two levels of a quantization scheme technique. The results for four standard square images exceeded eight times compression ratios compared to the original image with a good image quality. Ghadah et al. [15], adopted multi-resolution representation of two-level DWT, with all the details sub bands of the two layers quantized using the absolute block truncation coding (ABTC). The polynomial coding was applied to the second level approximation sub band, while the residual was first mapped to positive then sliced into its layers by applying bit plane slicing techniques (BPS). The least significant layers from layer 1 to layer 4 were ignored, while the most significant layers from layer 5 to layer 8 were quantized uniformly differently (each layer quantized with a scalar quantization step) and coded. The results were of high compression ratio with acceptable quality. Ghadah and Noor [16], utilized the one level decomposition residual based on DWT, with the hard or soft quantization process adopted for details sub bands, while the approximation sub band was quantized uniformly. The results showed the superiority of soft techniques for higher image quality compared to hard techniques for high compression ratios and lower quality. Ghadah and Sara [17], utilized the two-stage multiple description scalar quantizer (TSMDSQ) principle to efficiently quantize the residual image. The results are effective in terms of quality and compression performance. Ghadah [18], adopted the midtread adaptive quantizer to quantize the approximation sub band, along with soft quantization for the details sub bands, where the one level decomposition of DWT was used. Results were efficient and indicated high performance. Ghadah [19], utilized selected hard thresholding techniques of single or multiple base(s) to quantize the details sub-bands, while the approximation sub-band of one-layer DWT hierarchal scheme coded with the traditional linear polynomial coding. The results are of better performance compared to the traditional linear model where a higher compression ratio is achieved while preserving high image quality. Ghadah and Loay [20], introduced 1-D linear polynomial coding techniques that utilized two coefficients (a0, a1) for the deterministic part instead of the traditional model that used three coefficients (a0, a1, a2) for each segmented block, along incorporating a non-uniform quantization method for the probabilistic part (residual). Experimental results were promising in terms of performance (compression ratio, PSNR quality) for natural and medical grayscale images. Samara et al. [21] exploited the introduced 1-D polynomial coding techniques with matrix minimization algorithm of six values to efficiently compress residuals. The system achieved superior results than that adopted by Zhou et al. [22] using the same test images. The compression ratio was increased threefold compared to the first introduced 1-D scheme, with PSNR values converging to the compared mentioned work.

3. ADAPTIVE POLYNOMIAL CODING OF ITERATIVE BASED TECHNIQUES

As mentioned above, polynomial coding has been adopted by previous researches and can be considered as an extended revised version of predictive coding. This technique still suffers from residual and coefficients consumption, where actually the residual can be considered the main obstacle or difficulty compared to coefficients. In this paper we introduce a new method to efficiently represent polynomial coding of coefficients and residuals using an iterative based scheme. Figure 1 depicts the adaptive model, where the main contributions of the proposed system are:

1. This paper develops models for deterministic (coefficients) and probabilistic parts (residual).
2. It shows the effectiveness in terms of quality and compression ratios for spatial modelling techniques compared to the well-known standards techniques of JPEG and JPEG-2000.

The main steps of the algorithm are described as follows:

3. 1. Load the Original uncompressed image plane I of size $N\times N$, where I corresponds to an input image of N=256.

3. 2. Partition I Into Non-overlapping fixed sized blocks of size $n\times n$. The partition exploits the local dependency (correlation) embedded within image neighbourhoods, where no global correlation can be captured as a whole. In general, the fixed partition is utilized for simplicity without considering the homogeneity of blocks; the number of the fixed blocks equals to $(N/n)^2$, where here n = 4, so the number of blocks equals to $(256/4)^2 = 64\times64$ blocks.

3. 3. Compute the Coefficients of the linear polynomial coding according to Equations (1-4) [1, 6, 13, 15, 19], which implies three coefficients, where a_0 corresponds to the mean value of each block of size $n\times n$, a_1 and a_2 represent ratios of cumulative distances to both coordinates, and x_c, y_c correspond to the centre of the block.

$$a_0 = \frac{1}{n\times n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} I(i,j). \tag{1}$$

$$a_1 = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} I(i,j)\times(j-x_c)}{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(j-x_c)^2} \tag{2}$$

$$a_2 = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} I(i,j)\times(i-y_c)}{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(i-y_c)^2} \tag{3}$$

$$xc = yc = \frac{n-1}{2} \tag{4}$$

Here we have three arrays of the three computed coefficients each of size 64×64 blocks.

3. 4. Represent the Computed a_0 Coefficients of mean block values using iteration-based techniques. In other words, we introduce a new technique to model the a_0 coefficient values, which can be considered as adaptive

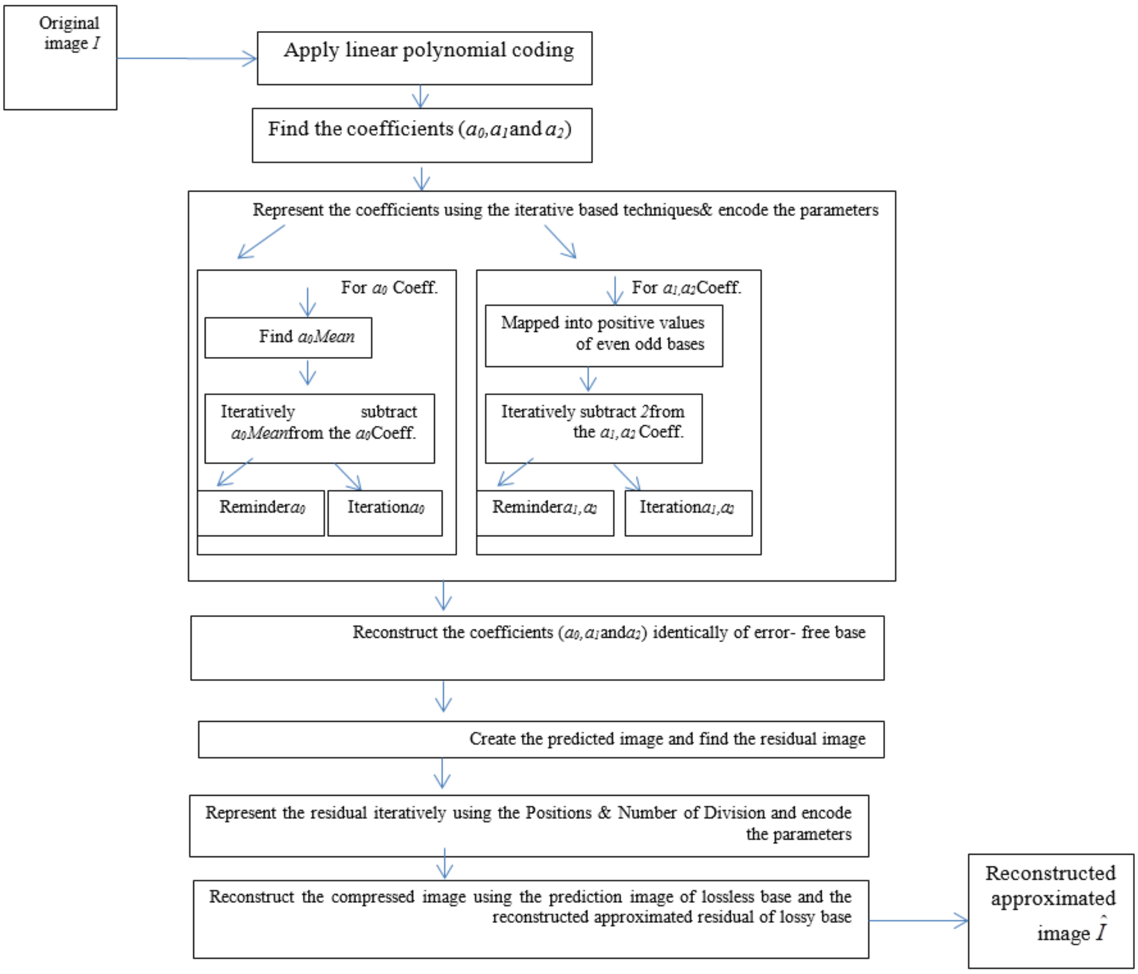


Figure 1. The proposed compression and decompression method

of the DPCM used in JPEG to encode the DC values, but with a recursive base of computed mean seed values. Put simply, start by computing the mean value of a_0 coefficients such as a_0M_{ean} according to Equation (5). Initially we compare each value in the a_0 coefficients array with the computed a_0M_{ean}: if the value is less than or equal to a_0M_{ean} then we keep the values as it is in Remainder with Iteration equal to zero, then for the values greater than the a_0M_{ean} we compare it recursively; namely for every iteration we subtract the mean value a_0M_{ean} from the a_0 coefficients with increments the iteration by one, until a_0 coefficient value becomes less than the threshold computed mean value a_0M_{ean}. Table 1 illustrates the steps using an example of one-dimension a_0 values with eight mean values; also, Algorithm (1) summarizes the techniques.

$$a_0M_{ean}(n,n) = \frac{1}{n \times n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_0(i, j) \qquad (5)$$

3. 5. Represent the Other Computed Coefficients (a_1 and a_2) effectively using the iteration principle, though here the scenario is different from a_0, since these values (a_1 and a_2) may be either negative or positive. Consequently, the first step is to map them into positive numbers of even and odd bases using Equation (6) [13].

$$Mapi_{Values} = \begin{cases} 2Coff_i & \text{if Coffi} \geq 0 \\ 2|Coffi| - 1 & \text{else} \end{cases} \qquad (6)$$

Here *Coff* corresponds to (a_1&a_2) values, $Mapi_{Values}$ mapped positive values, where positive values are mapped into even bases, while negative values are mapped into odd bases. Basically, the idea is to iteratively subtract a number – here we use base 2 since 2 is easily distinguished either even or odd base –from each value, which results in a binary representation of zeros and ones along the iteration number. It is important to remember to initially check these values in case the values are equal

TABLE 1. Example of a_0 recursive representation of Remainder and Iteration values, where the mean values of the eight a_0 values here equals to 62.

a_0 **original values**	**a0(1)**	**a0(2)**	**a0(3)**	**a0(4)**	**a0(5)**	**a0(6)**	**a0(7)**	**a0(8)**
	12	13	67	163	3	34	114	90
Remainder a_0	12	13	5	101 39	3	34	52	28
Iteration a_0	0	0	1	2	0	0	1	1

Algorithm (1). Recursive differencing a_0 coefficients encoding of mean-based techniques

Input: a_0 coefficient image of size $(N/n)^2$ (i.e., 64x64 for N=256, n=4)
Sm = 0; a_0M_{ean} = 0;
Output: Remaindera_0, Iterationa_0 each of size $(N/n)^2$ and a_0M_{ean}
Begin
//1- find size of a_0 image
[Rows, Cols] = size (a_0)
//2- calculate the mean (average) of a_0 image
for i = 1 : Rows
for j = 1 : Cols
Sm = Sm +a_0 *(i,j)*
End
End
a_0M_{ean}=*Sm/*(*Rows x Cols*)
//3- Initialize the two-output array (Remaindera_0 and Iterationa_0) each of size $(N/n)^2$ with values equal to zeros
Iteration$a_0(N/n)^2$ =0 , Remainder$a_0(N/n)^2$ *=0*
//4- Apply the proposed differencing technique
for i = 1 : Rows
for j = 1 : Cols
if $a_0(i,j)$*<=floor(*a_0M_{ean}*)* Remainder$a_0(i,j)$=$a_0(i,j)$, Iteration$a_0(i,j)$ =0.
if $a_0(i,j)$- a_0M_{ean} *>=1*
begin
If Remainder$a_0(i,j)$<=a_0M_{ean} Remainder$a_0(i,j)$= $a_0(i,j)$,Iteration$a_0(i,j)$ =Iteration$a_0(i,j)$+1.
Else $a_0(i,j)$=Remainder$a_0(i,j)$, Iteration$a_0(i,j$ =Iteration$a_0(i,j)$+1.
End if
End if
End if
End
End
End

to zeros or ones, with iteration number equal to zero. Table 2 illustrates an example of one-dimension a_1 values of eight mean values; also, Algorithm (2) summarizes the techniques.

3. 6. Encode/Decode the Compressed Information of coefficients representation (*Remainder* a_0 ,a_1, a_2, *Iteration* a_0, a_1, a_2) along the extra information (a_0Mean, *2*) using different coding techniques (Huffman coding/LZW) according to the parameter's nature.

3. 7. Reconstruct the Coefficients identically using the equations below, also illustrated in Tables 3 and 4:

$$a_0 = Remainder\ a_0 + (a_0Mean \times Iteration\ a_0) \quad (7)$$

$$a_1, a_2 = Remainder\ a_1, a_2 + (2 \times Iteration a_1 , a_2) \quad (8)$$

For the reconstructed coefficients of (a_1&a_2) bases the de-mapping process is required, such as described in literature [13]:

$$DeMap_i = \begin{cases} Rec(a_1, a_2)/2 & \text{if even} \\ -(Reca_1, a_2 + 1)/2 & \text{else} \end{cases} \quad (9)$$

where Reca$_1$,a_2 corresponds to reconstructed coefficients of even/odd bases.

3. 8. Create the Predicted Image $\tilde{I}$ using the original coefficient values of lossless base coding

TABLE 2. Example of a_1 recursive representation of Remainder and Iteration values.

a_1 **original values**	**a1(1)**	**a1(2)**	**a1(3)**	**a1 (4)**	**a1(5)**	**a1(6)**	**a1(7)**	**a1(8)**
	0	-3	-2	5	4	8	3	-1
a_1 **values after mapping**	0	5	3	10	8	16	6	1
Differencing		3 1	1	8 6 4 2 0	6 4 2 0	14 12 10 8 6 4 2 0	4 2 0	
Iteration a_1	**0**	**2**	**1**	**5**	**4**	**8**	**3**	**0**
Remainder a_1	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**

TABLE 3. Example of a_0 construed using the representation of Remainder and Iteration values along the mean

Remainder a_0	**12**	**13**	**5**	**39**	**3**	**34**	**52**	**28**
Iteration a_0	0	0	1	2	0	0	1	1
Use the encoded/decoded information using equation7								
***a_0* error-free reconstructed values**	12	13	67	163	3	34	114	90

TABLE 4. Example of a_1 construed using the representation of Remainder and Iteration values along the base of 2 value

Iteration a_1	**0**	**2**	**1**	**5**	**4**	**8**	**2**	**0**
Remainder a_1	0	1	1	0	0	0	1	1
Use the encoded/decoded information using equation8								
a_1 error-free reconstructed values before demapping	0	5	3	10	8	16	5	1
Use the encoded/decoded information using equation9								
***a_1*error-free reconstructed values after demapping**	0	-3	-2	5	4	8	-3	-1

Algorithm (2): Recursive differencing a_1, a_2 coefficients encoding proposed technique.

```
Input: a1,a2 coefficient images each of size (N/n)^2 (i.e., 64x64 for N=256, n=4)
Output: Remainder a1,a2, Iteration a1,a2 each of size (N/n)^2
Begin
//1- find size of a1 images
[Rows, Cols] = size (a1)
// 2- Mapped the values of a1,a2 images into even and odd values
 for i = 1 : Rows
   for j = 1 : Cols
     if (a1 (i,j) or a2 (i,j))>=0 Mapi_Values=(2x a1 (i,j)) or Mapi_Values= (2x a2 (i,j))
       else Mapi_Values=(2x abs(a1 (i,j))-1) or Mapi_Values=(2x abs(a2 (i,j))-1)
    End if
 End
End
//3- Initialize the two-output array (Remainder a1,a2 and Iteration a1,a2) each of size (N/n)^2 with values equal to zeros
Iteration a1,a2 (N/n)^2 =0 ,  Remainder a1,a2 (N/n)^2 =a0(i,j)
//4- Apply the proposed differencing technique
  for i = 1 : Rows
     for j = 1 : Cols
       If Mapi_Values(i,j) =0 orMapi_Values (i,j)=1     Iterationa1,a2=0 , Remaindera1,a2=Mapi_Values(i,j)
        if Mapi_Values (i,j)- 2>=1
        begin
         If Remainder a1,a2 (i,j)<=2 , Remainder a1,a2 (i,j)= a1,a2 (i,j), Iteration a1,a2 (i,j) =Iteration a1,a2 (i,j)+1.
         Else a1,a2 (i,j) =Remainder a1,a2 (i,j)  ,Iteration a1,a2 (i,j) =Iteration a1,a2 (i,j)+1.
         End if
       End if
     End if
   End
End
End
```

(Namely create the predicted image using the deterministic part), such as in literature [13,15]:

$$\widetilde{I} = a_0 + a_1(j - x_c) + a_2(i - y_c) \tag{10}$$

3. 9. Find the Residual (Difference) between original image I and the predicted one from the step above, this part corresponding to the probabilistic part in literature [13, 15]:

$$I\,\mathrm{Re}\,s(i, j) = I(i, j) - \widetilde{I}(i, j) \tag{11}$$

The residual is the vital part of the modelling process due to the prediction limitation (insufficiency) of capturing all the image characteristics using the same or various models for an image of varying details. Hence all the unpredicted information found in the residual which is essential for reconstructing the image, and in the same way is the core of the excessive bytes due to large uncorrelated data values that are difficult to manipulate directly, is traditionally solved using the lossy encoder of quantizer base, either of scalar base, which means the uniform/non-uniform techniques, or of vector base followed by a symbol encoder.

3. 10. Represent the Lossy Residual and iteratively using the scalar uniform base with predetermined thresholds of minimum and maximum values; this is necessary to preserve the quality of a minimum loss. In other words, each residual value is divided by 2 iteratively while it is within the quality range limited by maximum and minimum values. Each time, the remainder is kept with an increasing number of iterations. The main reason of using the value of 2 for division is the ability to exploit the values bit by bit (i.e., forcing the least significant bit to be the remainder until having forced all the other bits). Figure 2 illustrates an example of the residual iterative base; also, Algorithm (3) summarizes the techniques.

3. 11. Encode/Decode the Residual iterative representation, where the Number of Division parameter is coded using the popular Huffman coding, while the Position parameter which corresponds to the precision

Algorithm (3): Recursive division of residual based encoding techniques.

Input: *Residual* image of size (*N*×*N*) (*256×256*), *QuantizationFactor*=2
Output: Positions and Number of Division each of size (*N*×*N*)
Begin
//1- find size of a_1 images
[Rows, Cols] = size (Residual)
//2- Initialize the two-output array (Positions, Number of Division) each of size (N×N) with values equal to zeros values
Positions(*Rows*, *Cols*)=0 , Number of Division(*Rows*, *Cols*) =0

//3- Check if the residual values equals to zero
for i = 1 : Rows
for j = 1 : Cols
if (Residual (i,j) =0) Positions(i,j)=0 , Number of Division(i,j) =0
End if
End
End
Step 4:// Apply the proposed technique for non- zero residual values
While (all value in Residual not zero)
Matrix = Residual./QuantizationFactor; // Dot Division matrix by 2 ….
Iteration ++ ; // Increment iteration
If (Residual (i,j) >=MinimumQuality and <MaximumQuality)
Positions(i,j)=Residual , Number of Division(i,j) = Iteration
End if
End
End

matrix of floating-point values is subject to arithmetic coding. Our goal is to retain high accuracy with minimum degradation which is essential for conversion into integer number of preserving values, such as:

$$Positions = integer(Positions \times 10) \quad (12)$$

Here we convert the *Position* matrix into integer by keeping one significant digit after the decimal point. The integer position matrix is then coded using efficient arithmetic coding techniques.

3. 12. Reconstruct the Approximated Residual image values based on iterative lossy using the equations below. The coded data illustrated in Figure 2 is recovered and illustrated in Figure 3.

$$Positions = \frac{Positions}{10} \quad (13)$$

$$Values = 2^{Number\ of\ Divisions} \quad (14)$$

$$I\widehat{Res} = round(Values \times Positions) \quad (15)$$

3. 13. Rebuild the Compressed Image $\hat{I}$ by adding the approximated reconstructed residual image from the step above to the predicted the image from step 8, such as in [13, 15].

100 -64 - 12 78
23 24 65 90
34 76 56 -80
9 17 30 33
Iteration #0 (Original)

50.0 -32.0 -6.0 39.0
11.5 12.0 32.5 45.0
17.0 38.0 28.0 -40.0
4.5 8.5 15.0 16.5
Iteration #1 (Divide by 2)

25.0 -16.0 -3.0 19.5
5.75 6.0 16.25 22.5
8.5 19.0 14.0 -20.0
2.25 4.25 7.5 8.25
Iteration #2 (Divide by 2)

12.5 -8.0 -1.5 9.75
2.87 3.0 8.12 11.2
4.25 9.5 7.0 -10.0
1.12 2.12 3.75 4.12
Iteration #3 (Divide by 2)
Save RED values in matrix called Position (according to their X,Y)

6.25 -4.0 **0** 4.87
1.43 1.5 4.06 5.62
2.12 4.75 3.5 - 5.0
0 1.06 1.87 2.062
Iteration #4 (Divide by 2)
Save RED values in matrix called Position (according to their X,Y)

3.12 -2.0 **0** 2.43
0 **0** 2.03 2.81
1.06 2.37 1.75 -5
0 **0** **0** 1.03
Iteration #5 (Divide by 2)
Save RED values in matrix called Position (according to their X,Y)

1.56 -1.0 **0** 1.21
0 **0** 1.01 1.4
0 1.18 **0** -1.25
0 **0** **0** **0**
Iteration #6 (Divide by 2)
Save RED values in matrix called Position (according to their X,Y)

0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
Iteration #7 (Stop)

1.5 -1.0 -1.5 1.2
1.4 1.5 1.0 1.4
1.0 1.1 1.7 -1.2
1.1 1.0 1.8 1.0
Position Matrix
(corresponds to precision matrix of remainder base that is limited between maximum and minimum quality measures)

6 6 3 6
4 4 6 6
5 6 5 6
3 4 4 5
Number of Divisions
(comes from the number of iterations; at each stage when data are zero means stop counting for that data, replace it by iteration value)

Figure 2. Example of residual image block of size 4x4 with quality measures of maximum=2 and minimum=1

15 -10 -15 12	1.5 -1.0 -1.5 1.2	
14 15 10 14	1.4 1.5 1.0 1.4	
10 11 17 -12	1.0 1.1 1.7 -1.2	
11 10 18 10	1.1 1.0 1.8 1.0	
Converted into integer numbers by multiplying by 10	Original Position matrix (precision values of real numbers)	
64 64 8 64	96 -64 -12 77	100 -64 - 12 78
16 16 64 64	22 24 64 90	23 24 65 90
32 64 32 64	32 70 54 -77	34 76 56 -80
8 16 16 32	9 16 29 32	9 17 30 33
Values according to Equation (13)	Reconstructed residual values of minimum loss using the iterative based technique	Original residual values

Figure 3. Example of reconstructed residual image block of size 4x4 using the iterative lossy technique

4. EXPERIMENTAL RESULTS

In the experiments described here, we report on the amount of compression (number of bytes) compared using Huffman, Arithmetic Coding and the LZW-Lempel-Ziv-Welch algorithm. Concerning image quality, we use the objective fidelity criteria of PSNR (peak-signal to noise ratio) and NRMSE (normalized root mean squared error) (see Equations (17)-(18)), for simplicity, speed, and to facilitate comparisons with other related work. Test images of different types are shown in Figure 4. This includes natural, medical, and

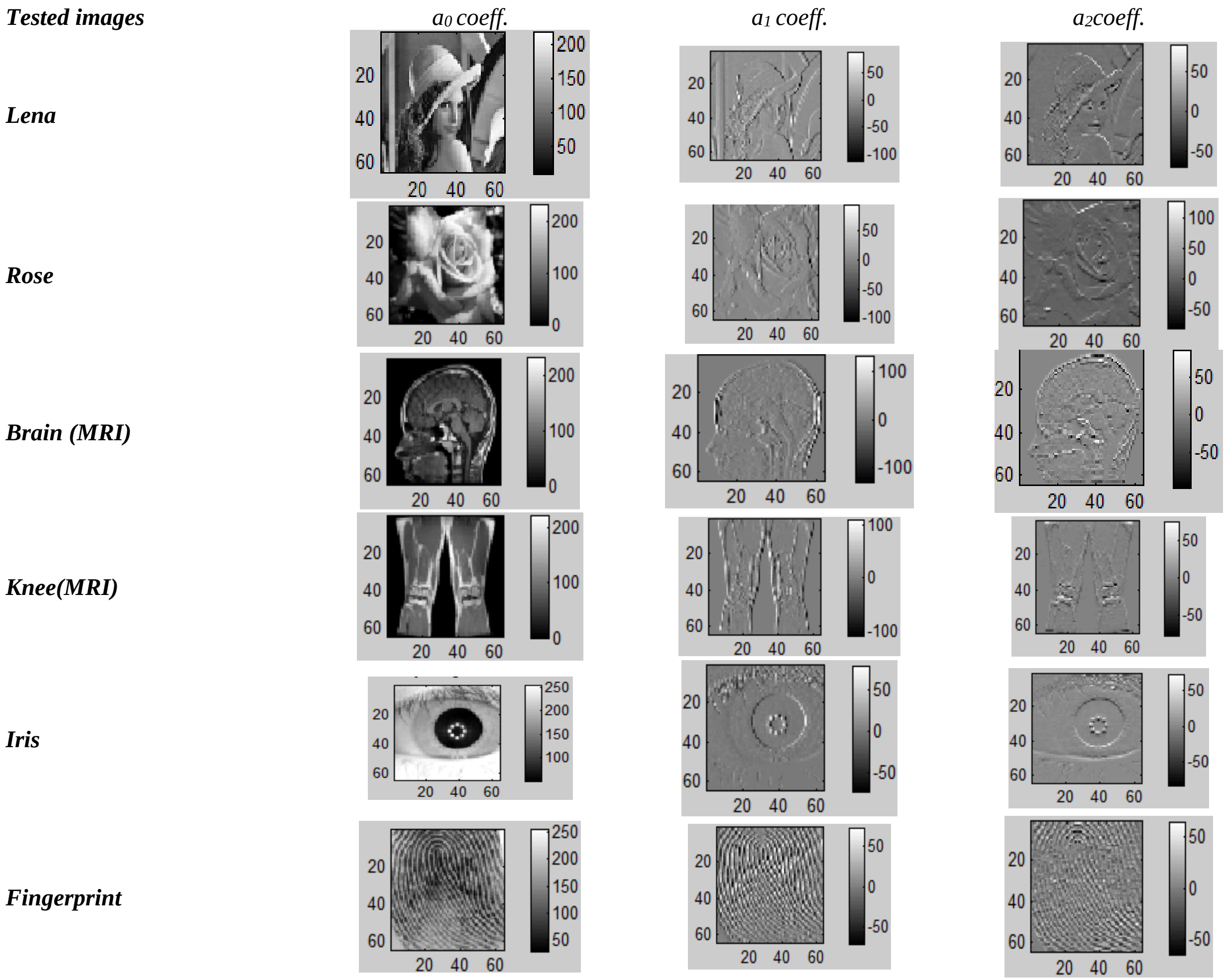


Figure 4. Test image coefficients (a_0, a_1, a_2) with range values

biometric images of varying details. All images are greyscale (8bits/pixels) of square size (256×256), and the block size used is 4×4. The proposed compression method was tested on a laptop computer with a processor Intel Corei 5-2450 CPU at 2.50GHz, 6 GB or RAM, using Matlab programming language. The fidelity measures defined as [1, 3-6]:

$$NRMSE(I,\hat{I}) = \sqrt{\frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[\hat{I}(x,y) - I(x,y)]^2}{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I(x,y)^2}} \quad (18)$$

$$PSNR(I,\hat{I}) = 10\log_{10}\left(\frac{(255)^2}{\frac{1}{N \times N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[\hat{I}(x,y) - I(x,y)]^2}\right) \quad (17)$$

where I represents the original uncompressed image and $\hat{I}$ represents the decoded compressed image.

4. 1. Experiment 1 The first experiment tested our proposed technique to lossless encoding polynomial coefficients (a_0, a_1, a_2), and comparing it to the traditional techniques of Huffman, arithmetic coding and LZW. Figure 5 shows the coefficients of the test images. Generally, for each of the coefficients (a_0, a_1, a_2) one byte was required (i.e., 64×64=4096 bytes for each coefficient). Tables 4 and 5 illustrate the size in bytes for the (a_0,a_1,a_2) coefficient values for the test images using the selected traditional techniques. In our proposed method, we use Huffman coding for iteration parameters and LZW for remainder parameters. This is because despite high repetition of iteration values meaning that

$$\hat{I}(x,y) = \tilde{I}(x,y) + I\widehat{Res}(x,y) \quad (16)$$

arithmetic coding would perform better than Huffman coding, the latter is simpler and, moreover, results showed that there are only small differences between them. Results clearly show that the proposed method has higher compression efficiency, which exceeds more than 2 times on average for all coefficient representations parameters. Tables 3, 4 and Figure 5 demonstrate the total number of bytes required for polynomial coefficients (a_0, a_1, a_2) using the Huffman coding and the adopted techniques. Figure 6 shows the performance comparison for the coefficients between the traditional coding techniques (Huffman coding, Arithmetic coding, LZW) and the proposed iterative techniques of error-free based.

4. 2. Experiment 2 Figure 7 shows the predicted and residual images of the test images. The second experiment results are shown in Tables 5-7 and Figure 8 which measuring the amount of residual image information before utilizing the representation of the iterative process of lossy base using the popular objective quantitative measure of root mean square error as follows [1]:

$$RMSE\,Res = \frac{1}{N^2}\sqrt{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I\,Res(x,y)^2} \quad (19)$$

The *RMSE Res* simply measures the amount of uncaptured image information due to the limitation of the prediction model which is directly affected by the image details or characteristic, around the edges of non-smooth details.

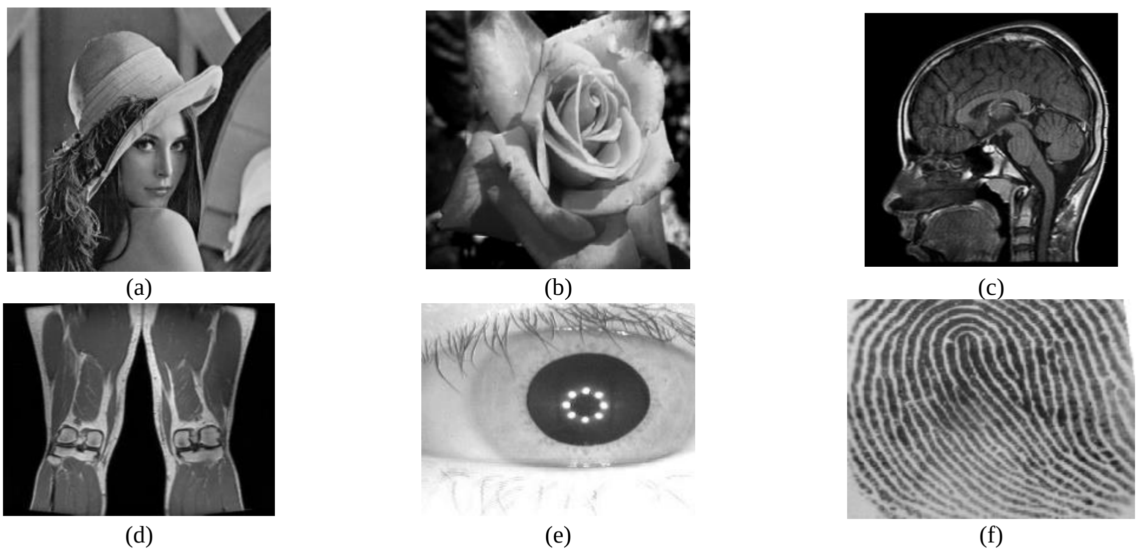


Figure 5. Test images are categorized into three groups, where (a) Lena and (b) Rose correspond to natural images, (c) Brain and (d) Knee correspond to medical images, and(e) Iris and (f) Fingerprint correspond to biometric images

TABLE 5. A comparison between coded techniques of *a1*, *a2* coefficients using traditional techniques and iterative base techniques

Tested Images	Coefficients	Number of bytes	Lossless encoding of the a_1, a_2 coefficients values			Lossless encoding of the a_1,a_2 coefficients for iteration based techniques		
			Huffman	Arithmetic coding	LZW	Remainder parameter	Iteration parameter	Total
Lena	a_1	4096	2796	2779	2612	863	490	1353
	a_2	4096	2392	2380	2152	790	348	1138
Rose	a_1	4096	2680	2668	2527	948	410	1358
	a_2	4096	2582	2566	2355	779	380	1159
Brain	a_1	4096	2566	2547	2337	986	358	1344
	a_2	4096	2462	2442	2350	840	352	1192
Knee	a_1	4096	2486	2462	2220	652	415	1067
	a_2	4096	2008	1978	1791	715	356	1071
Iris	a_1	4096	2286	2270	2256	862	466	1328
	a_2	4096	2268	2254	2012	866	356	1222
Fingerprint	a_1	4096	3440	3428	2987	1060	367	1427
	a_2	4096	3078	3064	2780	865	467	1332

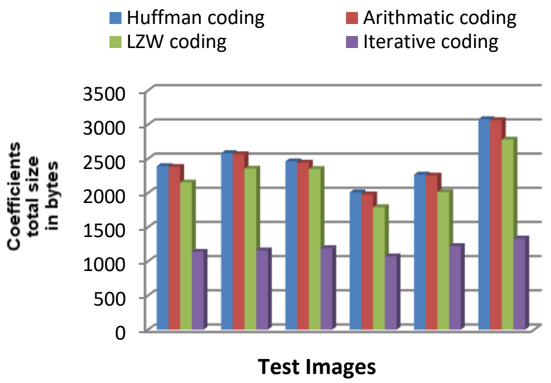


Figure 6. Comparison performance of the coefficients encoding techniques of traditional base (Huffman, arithmetic, LZW) and iterative base techniques

TABLE 6. Total number of bytes for the coefficients using the Huffman coding techniques and the proposed iterative based system for the test images

Tested images	Huffman coding	Proposed techniques
Lena	8958	3617
Rose	9186	3701
Brain	8136	3794
Knee	7660	3407
Iris	8112	3686
Fingerprint	10212	4113

4. 3. Experiment 3

This experiment is conducted to test how the parameters affects the residual iterative process of lossy base, namely the quality that is limited between maximum and minimum values. Here, three quality parameters were adopted that range between 1 and 2, 1 and 10, and 1 and 20, respectively. The PSNR (Equation (16)) between the original residual image and the reconstructed image was adopted, as shown in Table 8 and Figure 9(a) and (b). Additionally, SSIM measurement used to calculate the quality between residual image and the reconstructed image.

TABLE 7. The size of residual or prediction error for the tested images

Tested images	RMSE Res
Lena	12.0464
Rose	7.2657
Brain	14.1352
Knee	8.7756
Iris	8.6346
Fingerprint	13.4981

Certainly, the quality of residual images and byte consumption improves as the range of maximum and minimum values decrease; it is a trade-off between them, namely the higher the quality, the larger number of bytes related by a small range of values, and vice versa.

4. 4. Experiment 4

The last experiment was concerned with measuring the performance in terms of quality, compression time and compression ratio, which meant measuring the amount of encoded information in bytes which should be smaller than the original image.

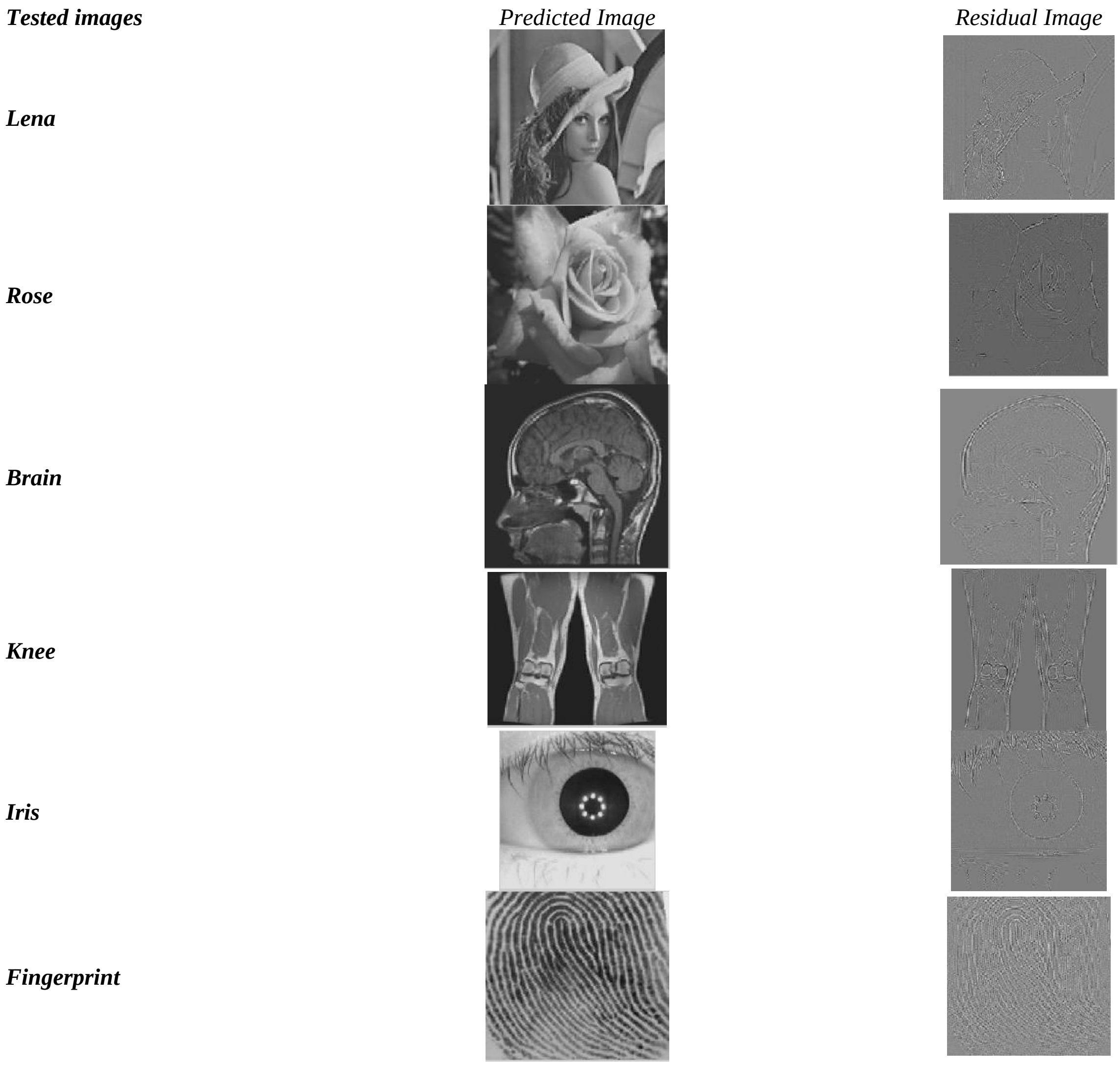


Figure 7. Tested prediction and residual images with block size of 4x4

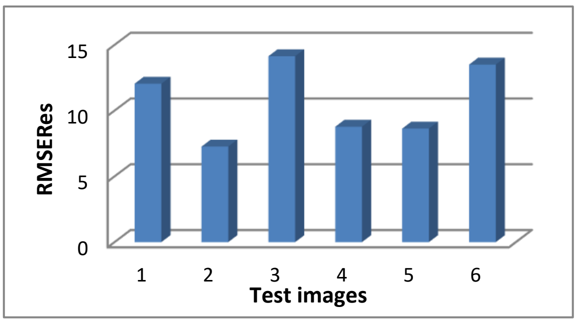


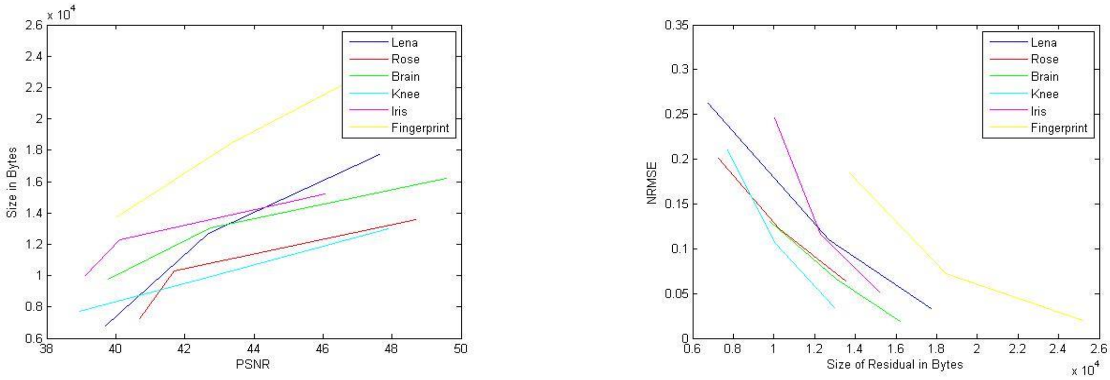
Figure 8. The amount of residual image information for each tested image in terms of RMSE

The compressed image size depends on the size of coefficients of lossless base and size of the residual of lossy base, along with the overhead information (a_0M_{ean}, base$_2$ for a_1, a_2 and the base$_2$ for division) of three extra bytes. So, the size of compressed information can be formulated such as in [1]:

$$Size_{Compressed} = Size_{Coefficients} + Size_{Residual} + Size_{ExtraInfo} \quad (20)$$

Table 9 and Figure 10(a) and (b) demonstrates the compression ratio versus the PSNR and NRMSE respectively for the tested images. Figure 11 shows the original and compressed tested images of high and low quality.

As expected, results showed an inverse relation between compression ratio and quality that is directly affected by the image details (characteristics) along with the effect of the quality residual measure minimum and maximum values. Also, the results illustrate that the total

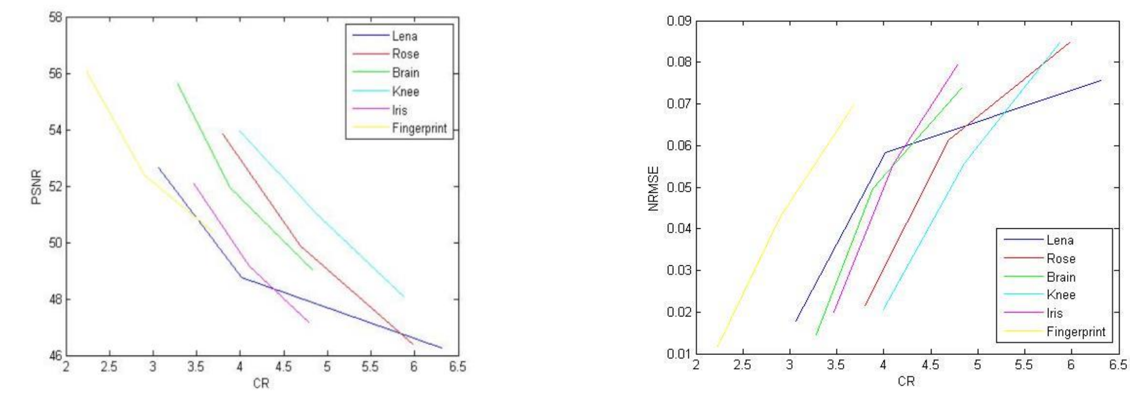


(a) (b)

Figure 9. Total size of the residual versus (a)PSNR and (b) NRMSE

TABLE 9. Compression performance for tested images

Tested Images	Coeff, in bytes	Limited by Quality		Position, in bytes	Total size, in bytes (eq. 20)	CR	PSNR (I, $\hat{I}$)	Quality NRMSE (I, $\hat{I}$)	SSIM (I, $\hat{I}$)	Total time in sec
		Min.	Max.							
		1	2	17749	21369	3.0669	52.6356	0.0178	0.972	7.0512
Lena	3617	1	10	12686	16306	4.0191	48.7623	0.0584	0.866	7.0356
		1	20	6768	10388	6.3088	46.2578	0.0755	0.892	6.9264
		1	2	13548	17252	3.7989	53.8307	0.0215	0.876	7.3164
Rose	3701	1	10	10279	13983	4.6869	49.8702	0.0614	0.811	7.1760
		1	20	7254	10958	5.9807	46.3887	0.0847	0.833	6.8660
		1	2	16195	19992	3.2781	55.6287	0.0145	0.931	5.9436
Brain	3794	1	10	13071	16868	3.8852	51.9552	0.0496	0.895	5.8344
		1	20	9798	13595	4.8206	49.0311	0.0737	0.877	5.7865
		1	2	12984	16394	3.9976	53.9435	0.0207	0.934	6.1308
Knee	3407	1	10	10087	13497	4.8556	51.0609	0.0557	0.953	6.0020
		1	20	7749	11159	5.8729	48.0721	0.0848	0.833	5.9804
		1	2	15211	18900	3.4676	52.0969	0.0199	0.812	6.9732
Iris	3686	1	10	12304	15993	4.0978	49.1702	0.0555	0.864	6.8640
		1	20	10020	13709	4.7805	47.1879	0.07933	0.798	6.6371
		1	2	25208	29324	2.2349	56.0829	0.0118	0.941	6.1528
Fingerprint	4113	1	10	18476	22592	2.9009	52.3889	0.0429	0.953	6.0996
		1	20	13722	17838	3.6739	50.4090	0.0695	0.875	5.9592



(a) (b)

Figure 10. Compression ratio versus the (a) PSNR and (b) NRMSE for the tested images

(a) (b) (c)

Figure 11. Other tested natural images, where (a) Card and (b) Apple correspond to natural images, and (c) Guitar images. Each image is 1200x1200 pixels, 1.37MB.

compression time- encoding of iterative based techniques and direct decoding process – is inversely related to the range of the residual quality measures; a small range has a large number of division iterations, and as the range increases the division iteration numbers decrease, with decreasing time. The interesting point is the excellent near perfect quality of the decoded compressed images. It is subjectively impossible to differentiate between the compressed image and the original one. This is due to preserving image information in terms of lossless coefficients causing minimum degradation or minimum residual loss.

Finally, the comparison with the well-known standard techniques JPEG and JPEG2000 is given in Table 10, based on measuring the compression ratio and the quality in terms of PSNR for the test images shown in Figure 4. Also, other test natural images added for comparative analysis of performance are shown in Figure 12. They follow the same criteria adopted for the previous images, namely they are greyscale square images of size (256×256). Figures 13 and 14 show a direct comparison of JPEG and JPEG-2000 set at the highest image quality with our technique compressed at lower quality. The decoded images in JPEG/JPEG2000

TABLE 10. PSNR of JPEG set on the highest quality compared to the original image.

Tested Images	JPEG				JPEG-2000			
	Total size in bytes	CR	PSNR	SSIM	Total size in bytes	CR	PSNR	SSIM
Lena	11366	5.7659	38.8708	0.761	10879	6.0240	41.3328	0.901
Rose	10762	6.0895	41.0337	0.721	8704	7.5294	43.7361	0.987
Brain	11858	5.5267	39.8728	0.812	10137	6.4650	42.4219	0.954
Knee	9728	6.7394	41.2240	0.952	9113	7.1914	45.0710	0.899
Iris	8908	7.3567	40.3316	0.912	10235	6.4031	43.9751	0.879
Fingerprint	15698	4.1747	38.8799	0.912	11035	5.9389	40.1820	0.946
Card	14336	4.5614	34.5320	0.871	10822	6.0558	36.7908	0.932
Apple	13207	4.9622	41.0911	0.911	11666	5.6176	44.8534	0.923
Guitar	11288	5.8085	39.8915	0.991	9830	6.6669	43.1997	0.988

Test Image	*Original image*	*Compressed at higher quality*	*Compressed at lower quality*
Lena			
Rose			

Brain

Knee

Iris

Fingerprint

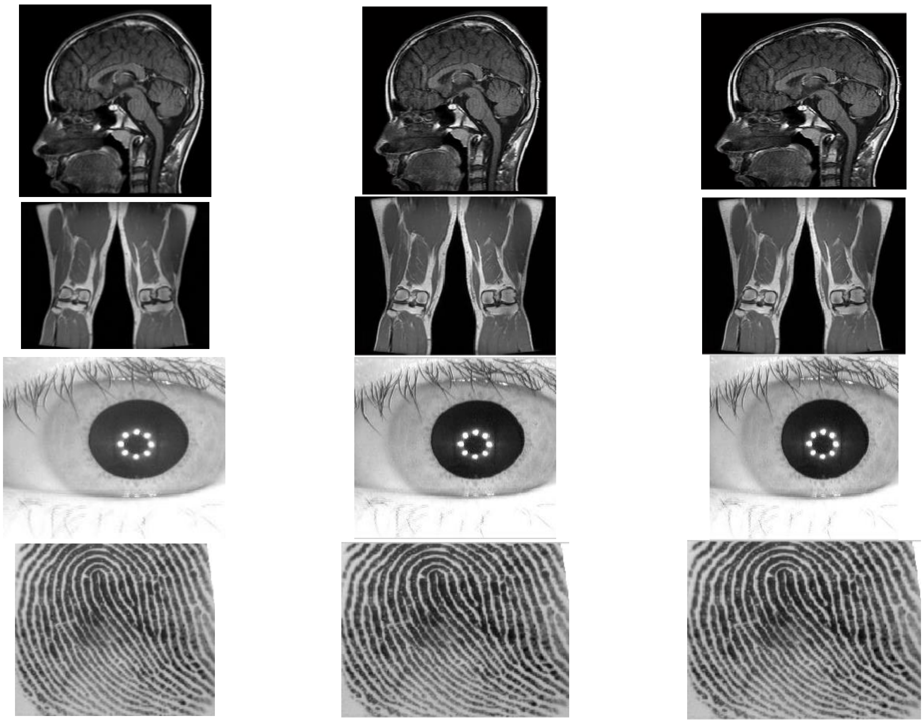


Figure 12. Examples of original test images and compressed images of high/low quality by our proposed method. Each image is 256x256 pixels, 65 KB

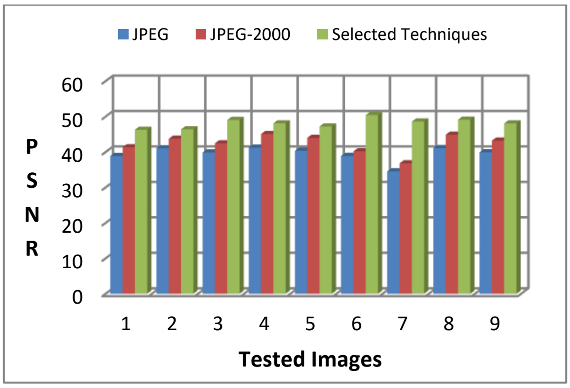


Figure 13. PSNR of JPEG/JPEG2000 versus the proposed technique for the tested images

are inferior to our method, even when our method is set to low quality (to yield similar compression ratios as JPEG/JPEG2000). Therefore, it is demonstrated the superior performance of our method with higher PSNR values as compared to JPEG/JPEG2000, for similar compression ratios. Also, the other comparison performed with traditional polynomial and two adaptive works relied on the Lena/Rose test images is given in Tables 11 and 12; where superior higher quality is achieved compared to litrature. Even with high compression ratios performed , still our results are promising with a clear trade-off between quality and compression ratio.

Image ***JPEG reconstructed*** ***JPEG-2000 reconstructed*** ***Our method set at low image quality***

Lena

Rose

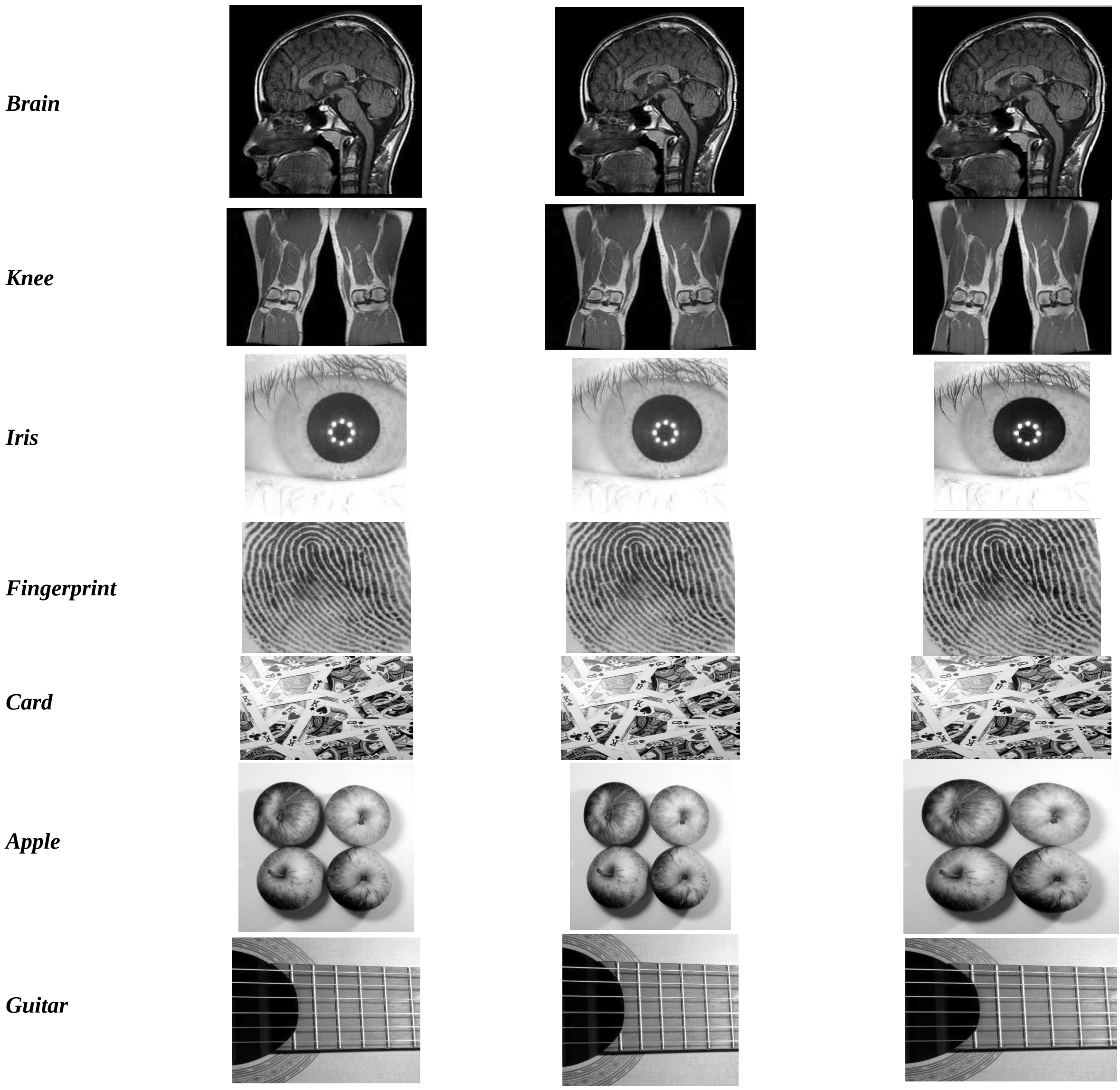


Figure 14. Examples of original tested images and compressed images of JPEG technique set at the highest quality and the suggested technique set at low quality

TABLE 11. Comparison with traditional polynomial, adaptive techniques and the proposed system for Lena image

Image Compression Techniques of traditional coding, adaptive coding and the proposed	**Performance for Lena Tested image**		
	CR	**PSNR**	**SSIM**
Traditional polynomial coding block size *4x4*, Quantization Coeff.*1,2,2*, and Quantization Res *5*	3.3227	45.0201	0.889
Traditional polynomial coding of 2D base, block size *4x4*, Quantization Coeff.*1,2,2*, and Quantization Res *40*	4.4329	31.1426	0.432
adaptive polynomial coding of 2D hard thresholding base, block size *4x4*, Quantization Coeff.*1,2,2*, and thresholding of subbans coding *20,20,40* and approximation subband *2* [23]	5.1312	29.9972	0.219
adaptive polynomial coding of 2D soft thresholding base, block size *4x4*, Quantization Coeff.*1,2,2*, and thresholding of subbans coding *20,20,40* and approximation subband *2* [23]	4.9201	33.3726	0.495
Adaptive polynomial with Quantization Steps of Coefficients are *1,2,2, LHThr=21,HLThr=36,HHThr=32,* Using the Seven Midtread Quantization base adopted by Burget & Das that utilized the minimum standard deviation value of residual image	8.5556	31.7175	0.456
Proposed system with quality between *1*to *2*	3.0669	52.6356	0.972
Proposed system with quality between *1*to *10*	6.3088	46.2578	0.892

TABLE 12. Comparison with traditional polynomial, adaptive techniques and the proposed system for Rose tested image

Image Compression Techniques of traditional coding, adaptive coding and the proposed	Performance for Rose Tested image		
	CR	PSNR	SSIM
Traditional polynomial coding block size *4x4*, Quantization Coeff.*1,2,2,* and Quantization Res *5*	3.7186	45.4949	0.828
Traditional polynomial coding of 2D base, block size *4x4*, Quantization Coeff.*1,2,2,* and Quantization Res *40*	4.4783	33.2660	0.638
Adaptive polynomial with Quantization Steps of Coefficients are *1,2,2, LHThr=21, HLThr=36, HHThr=32,* Using the Seven Midtread Quantization base adopted by Burget & Das that utilized the minimum standard deviation value of residual image	9.6718	35.5568	0.532
Proposed system with quality between *1*to *2*	3.7989	53.8307	0.876
Proposed system with quality between *1*to *10*	5.9807	46.3887	0.833

5. CONCLUSION

This paper proposed a novel iterative image coding technique based on an efficient hybrid lossy technique. The significance of our proposed methods is that they are convenient for a variety of image types including natural, medical and biometric grey level images. For the latter two types compression is critical, and is normally coded in lossless manner (error-free) as priority is given to keeping all information from the image. The experiments shown here demonstrate our proposed technique to a wide range of images where the quality of all tested images in terms of PSNR exceeds the well-known standard techniques of JPEG and JPEG-2000.

The iterative part constitutes the core of the paper and uses two different schemes, a lossless followed by a lossy method. First, the lossless method is based on a set of polynomial coefficients a_0 and (a_1,a_2) where a_0 is characterized by efficiently embedding correlations by subtracting the mean value at each iteration and keeping the number of iterations with the remainder. The mapping/de-mapping process is essential for converting the coefficients (a_1,a_2) values from negative and positive values into even/odd base to overcome the sign problem of negative numbers which requires a large number of bytes. The iterative process applies base2 differential techniques with superior representational performance converting uncorrelated, large byte consuming values into efficient representation of number of iterations and remainder parameters. Second, the lossy method is based on the residual that represents the number of divisions along the remainder. It is used to reconstruct an approximated value with minimum loss controlled by a maximum and minimum quality range that resembles the non-uniform quantization process.

The considerations above highlight the main limitations of our proposed method in relation to complexity, which may represent obstacles to its wide use. The average time complexity of the methods is estimated as O (n log n). Before the methods can be widely adopted (at par with other techniques such JPEG/JPEG2000) the following aspects are required to be addressed:

1. Standardization/practical issues: the proposed system produces high quality images with good compression ratios, but is still complex and needs to be optimized.
2. Performance issues: the polynomial coding is promising and simple to implement, however, there are a number of related issues that need to be developed further:

- The simplicity of the utilized symbol encoder techniques.
- Extending the system to utilize a hybrid system of the transform coding, by incorporating frequency techniques such as discrete wavelet transforms (DWT) or discrete cosine transform (DCT).
- Extending the system by mixing between the linear and the non-linear polynomial based techniques allowing the block nature to efficiently reduce the residual.
- Exploiting the region of interest (ROI) based segmentation process, especially in medical or frontal face images, to use the lossy background effectively.

3. Extending the proposed system to work with colour images; an initial solution could be simply repeat the method for each image plane.

Research on the above issues is under investigation and results will be reported in related works

6. REFERENCES

1. Al-Khafaji, G., "Intra and inter frame compression for video streaming", ***Computer Science***, (2012).
2. Shi, Y.Q. and Sun, H., "Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards, CRC press, (1999).
3. Gonzales, R.C. and Woods, R.E., "Digital image processing, 2-nd edition", Prentice Hall, New Jersey, 2nd edition January, Vol. 15, No. 2002, (2002), 793.

4. Ghanbari, M., "Standard codecs: Image compression to advanced video coding, Iet, (2003).
5. Sayood, K., "Introduction to data compression, Morgan Kaufmann, (2017).
6. Kaur, M. and Kaur, G., "A survey of lossless and lossy image compression techniques", ***International Journal of Advanced Research in Computer Science and Software Engineering***, Vol. 3, No. 2, (2013), 323-326. https://doi.org/10.23851/mjs.v29i2.242
7. Setyaningsih, E. and Harjoko, A., "Survey of hybrid image compression techniques", ***International Journal of Electrical and Computer Engineering***, Vol. 7, No. 4, (2017), 2206. doi: 10.11591/ijece.v7i4.pp2206-2214.
8. Hussain, A.J., Al-Fayadh, A. and Radi, N., "Image compression techniques: A survey in lossless and lossy algorithms", ***Neurocomputing***, Vol. 300, (2018), 44-69. doi: 10.1016/j. neucom.2018.02.094.
9. Rahman, M.A. and Hamada, M., "Lossless image compression techniques: A state-of-the-art survey", ***Symmetry***, Vol. 11, No. 10, (2019), 1274. doi: 10.3390/sym11101274.
10. Kotha, H.D., Tummanapally, M. and Upadhyay, V.K., "Review on lossless compression techniques", in Journal of physics: conference series, IOP Publishing. Vol. 1228, (2019), 012007.
11. Siddeq, M.M. and Al-Khafaji, G., "Applied minimized matrix size algorithm on the transformed images by dct and dwt used for image compression", ***International Journal of Computer Applications***, Vol. 70, No. 15, (2013). doi: 10.5120/12040-8000.
12. Siddeq, M. and Rodrigues, M., "A new 2d image compression technique for 3d surface reconstruction", International Conference on Circuits, Systems, Communications and Computers, Santorin Island, Greece, (2014), 379-386.
13. George, L.E. and Sultan, B., "Image compression based on wavelet, polynomial and quadtree", ***Journal of Applied Computer Science & Mathematics***, Vol. 11, No. 5, (2011), 15-20. doi: 10.5120/13229-0658.
14. George, L. and Dhannon, B., "Image compression using polynomial and quadtree coding techniques", ***International Journal of Scientific & Engineering Research***, Vol. 4, No. 11, (2013), 2229-5518. https://doi.org/10.47577/technium.v4i6.7008
15. Rasha, A.-T., "Image compression using enhancement polynomial prediction coding", Master thesis, University of Baghdad, Iraq, (2015),
16. Al-Khafaji, G. and Al-Mahmood, H., "Lossless compression of medical images using multiresolution polynomial approximation model", ***International Journal of Computer Applications***, Vol. 76, No. 3, (2013).
17. Al-Khafaji¹, G. and Bassim, M., "Color image compression of inter-prediction base", ***International Journal of Computer Science and Mobile Computing***, Vol. 8, No. 11, (2019), 65-70.
18. Al-Khafaji, G. and Al-Kazaz, H.B., "Adaptive color image compression of hybrid coding and inter-differentiation based techniques", ***International Journal of Computer Science and Mobile Computing*** Vol. 8, No. 11, (2019), 65-70.
19. Al-Khafaji, G., "Image compression based on quadtree and polynomial", ***International Journal of Computer Applications***, Vol. 76, No. 3, (2013). doi: 10.5120/13229-0658.
20. Al-Khafaji, G. and George, L., "Grey-level image compression using 1-d polynomial and hybrid encoding technique", ***Journal of Engineering Science and Technology***, Vol. 16, No. 6, (2021), 4707-4728. doi. https://jestec.taylors.edu.my/V16Issue6.htm
21. AL-Hadithy, S.S., "Adaptive 1-d polynomial coding of c621 base for image compression", ***Turkish Journal of Computer and Mathematics Education (TURCOMAT)***, Vol. 12, No. 13, (2021), 5720-5731. https://www.turcomat.org/index.php/turkbilmat/article/view/9823
22. Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., "Image quality assessment: From error visibility to structural similarity", ***IEEE Transactions on Image Processing***, Vol. 13, No. 4, (2004), 600-612. doi: 10.1109/TMM.2011.2152382.
23. Al-Khafaji, G. and Mahdi, N.S., "Image compression based on adaptive polynomial coding of hard & soft thresholding", ***Iraqi Journal of Science***, Vol. 57, No. 2B, (2016), 1302-1307. https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/7392

Persian Abstract

چکیده

با افزایش تقاضا برای استفاده فشرده از تصاویر، به ویژه مرتبط با برنامه های کاربردی آنلاین و همچنین انقلاب عظیم و مداوم فناوری تلفن همراه، نیاز به تکنیک های فشرده سازی کارآمد و استاندارد که سادگی و سرعت را تضمین می کند، پدیدار شده است. اینها باید با نیازهای کاربر سازگار باشند، اما با چالش‌های بهبود تکنیک‌های فشرده‌سازی نیز مواجه شوند. کدگذاری چند جمله‌ای یکی از این تکنیک‌ها است که هنوز در حال توسعه است، بر اساس مفهوم مدل‌سازی مبانی کدگذاری قطعی و احتمالی. این مقاله یک مدل چند جمله ای تکراری ریاضی جدید را برای نشان دادن هر دو پایه کدگذاری معرفی می کند. این مدل یک روش ترکیبی کارآمد را پیشنهاد می‌کند که در آن ضرایب به‌عنوان بدون تلفات نشان داده می‌شوند در حالی که باقیمانده‌ها به‌عنوان تلفات اما با حداقل تلفات ارائه می‌شوند، که عملکرد مؤثر را از نظر نسبت تراکم و کیفیت تضمین می‌کند. نتایج نشان می‌دهد که در حالی که این تکنیک دارای محدودیت‌هایی است، سیستم پیشنهادی به نسبت‌های فشرده‌سازی معادل تکنیک JPEG استاندارد دست می‌یابد، اما با کیفیت برتر برای نسبت فشرده‌سازی مشابه.