DOI: 10.24996/ijs.2024.65.6.36

ISSN: 0067-2904

Gray-Scale Image Compression Method Based on a Pixel-Based Adaptive Technique

Zahraa.H. Abed*, Ghadah K. AL-Khafaji
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Received: 20/12/2022 Accepted: 6/4/2023 Published: 30/6/2024

Abstract
Today in the digital realm, where images constitute the massive resource of the social media base but unfortunately suffer from two issues of size and transmission, compression is the ideal solution. Pixel base techniques are one of the modern spatially optimized modeling techniques of deterministic and probabilistic bases that imply mean, index, and residual. This paper introduces adaptive pixel-based coding techniques for the probabilistic part of a lossy scheme by incorporating the MMSA of the C321 base along with the utilization of the deterministic part losslessly. The tested results achieved higher size reduction performance compared to the traditional pixel-based techniques and the standard JPEG by about 40% and 50%, respectively, with pleasing quality exceeding 45 dB.

Keywords: Image Compression, Matrix Minimization Search Algorithm, Pixel-Based, JPEG, Limited Space Search Table

طريقة مطورة لضغط الصور الرمادية القائمة على البكسل

زهراء حسام عبد*، غادة كاظم الخفاجي
قسم علوم الحاسوب ، كلية العلوم ، جامعة بغداد ، بغداد ، العراق

الخلاصة
اليوم في العالم الرقمي ، حيث تشكل الصور مورد اساسي لشبكة التواصل الاجتماعي ، لكنها للأسف تعاني من مشكلتين من حيث الحجم والإرسال ، لذا فإن الضغط هو الحل الأمثل. تعد تقنيات القائمة على البكسل احدى تقنيات النمذجة المكانية الحديثة تتألف من جزأين (deterministic part and probabilistic part)والتي تتضمن (mean, index and residual) هذا البحث يقدم تقنيات محسنة لتقنية القائمة على البكسل وخصوصا (probabilistic part) وبالاعتماد على نموذج الفقدان من خلال استخدام Matrix Minimization Search Algorithm(MMSA) of c321 بالإضافة الى (deterministic part) بدون فقدان . النتائج التجريبية بينت أداءً أعلى في تقليل الحجم مقارنةً بالتقنيات التقليدية القائمة على البكسل و JPEG القياسي بحوالي (40٪) و (40–50٪) على التوالي ، مع جودة مرضية تتجاوز 45dB.

1. Introduction

Today, the number of people that are active online exceeds 2.5 billion, and they usually utilize digital devices (smart phones, computers, iPads, and smart watches) and digital products of instant messaging (e.g., Viber, WhatsApp) and social media (e.g., Facebook,

*Email: zahraahusam83@gmail.com

Twitter, Instagram) to facilitate people's contacts cheaply and safely, especially in the case of the Corona virus, where the multimedia realm encompasses image, video, audio, and text [1, 2]. The image represents the cornerstone or backbone of all daily life applications [3, 4].

Image compression systems are concerned with reducing bytes consumption by preserving image information compactly while eliminating redundancy that implies between neighboring pixels, using fixed length coding representations, and exploiting the limitations of the Human Visual System (HVS) perception [5, 6, 7]. Therefore, the main goal of image compression is to represent the image in the fewest possible bits without losing the content of the necessary and basic information within the scope of the original image [8, 9, 10]. Compression techniques have been developed to overcome and address the challenges and problems that have emerged recently in view of the increasing growth of technology [11, 12], where huge amounts of data must be stored correctly using compression algorithms in different ways [13]. Basically, the image compression techniques are categorized according to the way of redundancy removal or exploitation, which implies two classes: lossless and lossy [14].

Pixel-Based Technique (PBT) is a new simple and efficient model to eliminate the spatial (interpixel) redundancy by identifying the model of each pixel value that corresponds to the best value of the minimum residual [15]. The suggested technique works to compress a grayscale square image of size N×N. An algorithm (1) shows the full steps for PBT. This paper presented an improvement to PBT for grayscale image compression, which is based on pixel modeling and the exploitation of minimal residual, relying on MMSA (which losslessly represents compressing every three data points to a single floating point value by exploiting weighted random values between [0] and [1]), three-to-one (c321) data compression, and other efficient compression techniques [16, 17]. The proposed system proved its high performance in terms of reducing the size while maintaining a pleasing quality by dividing the residual into two parts, the Most Significant Value (MSV) and the Least Significant Value (LSV) [18]. The MSV part depends on the use of MMSA and the application of LZW and Huffman methods. The part of the LSV that contains the least important information uses uniform scalar quantization followed by wavelet transform (DWT) in two levels and applies a hard threshold to only three sub-bands (LH, HL, HH), followed by LZW and Huffman. The paper is organized as follows: The second section is concerned with the most relevant work, and the third section discusses the proposed system. The rest of the sections are concerned with experimental results and conclusions.

2- Related work

This study can be divided into two parts: the first section is a survey dedicated to various pixel-based techniques (PBT), and the second section contains previous works in which the Matrix Minimization Size Algorithm (MMSA) was used.

- Eduardo et al. [19] suggested a lossless compression system. Two types based on pixels were used to predict using arithmetic coding and lossless compression of grayscale images. The first model included the use of two prediction forms. The first included the pixel intensity in the upper left neighborhood and upper right neighborhood (UL/UR), while the other was the right (R). the Left (L), the Bottom (B), and the Upper (U), while the second model was represented by three shapes that were used to predict one or four neighbors. The other used either UL, UR, Left Bottom Neighbor (LB), Bottom Left Neighbor (BR), or U, L, R, B, where the former used the left bottom. The system was tested by the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) T.24 for the dataset on three known images of medical, physical, and photo text/drawing grayscale, and the CR results ranged between 1.240 and 4.99 for the first model, while the CR results for the second model ranged from 1.244 to 5.650, as the results vary according to the image complexity of the matched replayed image base.

- Abdullah and Ghadah [15] proposed a hybrid system to compress grayscale images, where a new technique based on pixels was proposed to reduce the size of the residual. Haar DWT was used with hierarchal quantization. The Haar DWT is dependent on three quantization parameters (Q, β, and α) that affect the system's performance. Pressure was higher for higher values of these parameters (approximately 15%). Three standard square grayscale images of size 256 × 256 were used. Each image has different neighbors (ngb) and increment values (inc), and CR values of more than 14 were obtained in addition to maintaining a quality of more than 41 dB.
- Anitha et al. [20] presented a lossy compression system for grayscale images using an intra-prediction algorithm of video compression standards. The system is based on partitioning the input image into fixed square block sizes from 4×4 up to 32×32 pixels, then, for each current block, calculating Block Distortion Measure (BDM) for the eight neighboring blocks in the square range. If the minimum BDM is zero in any of the neighboring blocks, the predicted experimental findings were tested on seven 256×256/512×512 grayscale square images, with average CR and PSNR of 5.6325 and 56.7008 dB, respectively.

The second section is dedicated to discussing various Minimize Matrix Size Algorithm (MMSA) techniques, such as:

- Mohammed et al. [21] presented a lossy compression system to compress color, grayscale, and structured light images using a mixture of the discrete Fourier transform (DFT) and the matrix minimization (MM) algorithm. First, the original image is divided into blocks (4 × 4) that are not overlapping. After that, DFT is applied to each block independently, and two parts are produced: the real (low-frequency coefficients and high-frequency coefficients) and the imaginary. After that, uniform quantization is applied to each part. Finally, the MM algorithm is applied to each part. The compression process involves multiplying three coefficients into three random keys (each value multiplied by a different random key) and then summing them up. In the decompression stage, the compressed coefficients are retrieved to their original three values by searching between the MIN and MAX values in order to speed up the recovery process.
- Samara et al. [17] proposed adaptive 1-D polynomial coding techniques with a matrix minimization algorithm of six-to-one data (C621) for the grayscale image to compress (the residual part). The experimental results were tested on six standard gray square images of medical and natural backgrounds and compared to traditional 1-D coding techniques. The compression ratio increased by more than three times compared to traditional 1-D and PSNR values that are convergent to the referred work.

3- Proposed system

The technique for modeling that has been developed uses the correlation between neighboring pixels. Because neighboring pixels are not statistically independent, it is possible to take advantage of this dependency between neighboring pixels and construct a mathematical model based on determining the mean value for each pair (row) of correlating pixels. The proposed compression system consists of the following basic steps, and Figure 1 shows the steps of the proposed system structure.

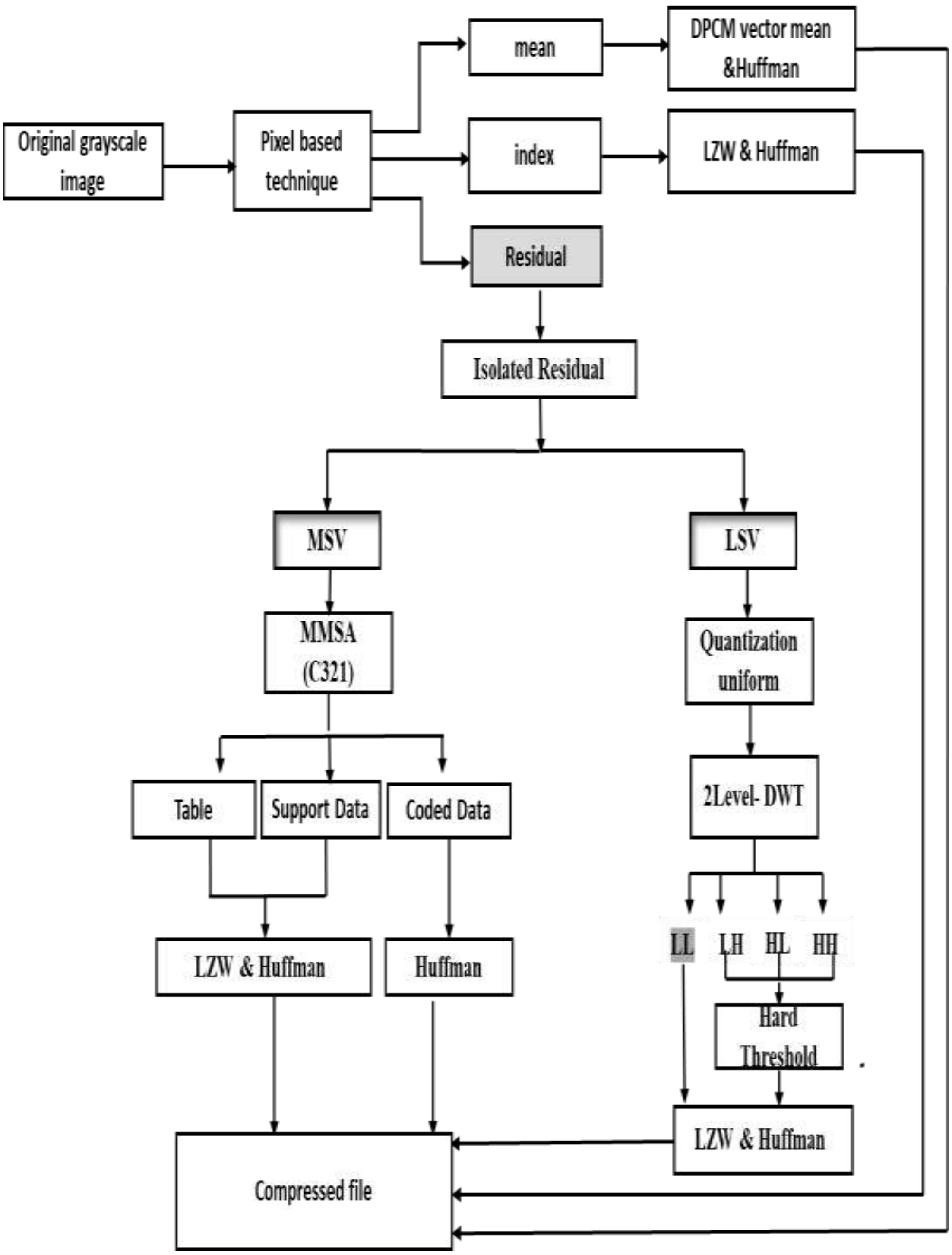


Figure 1: Structure of proposed System

Algorithm (1): Pixel Based Coding Techniques	
Input	J // gray scale image nbr , SP // no.of neighbors (nbr), increment value(SP)
Output	res, mean, location // residual array(res), mean vector (mean)and indexes array(location)
Begin End	**Step 1**:Read input grayscale image (J) , number of neighbor (nbr) & increment value (SP) **Step 2:** //find mean value for each row by equations (1) followed by Huffman **Step 3**: //Calculate neighbor vector (Nvector) per mean value **Step4 :// C**alculate indexes array by equation(3) and apply LZW algorithm **Step 5**: For residual part, isolate MSV from LSV then code the former using MMSA, while the latter using 3 level DWT followed by hard thresholding techniques of details subbands. followed by LZW & Huffman.

Step 1: Read the original gray image *J* of size N×N of BMP format.
Step 2: Calculate the mean for each row in the image by reading row by row, as shown in Table 1, and such as:

$$Mean(M) = \frac{1}{N}\sum_{i}^{N} J(N,N) \quad (1)$$

Table 1: An example of the computed mean value of the first row in an image is 90.

Values	***V1***	***V2***	***V3***	***V4***	***V5***	***V6***	***V7***	***V8***	***V9***	***V10***
First row values	185	160	135	120	60	60	40	40	50	50

Where *V* corresponds to the pixel value
Step4: Choose two control parameters, the first of which corresponds to the number of neighbors (nbr), and the second of which corresponds to the increment value between the pixels (SP), to create a neighbor vector by determining the number of nbr and SP, as shown in Eq. (2) and as shown in Table 2.

$$Nvector(r) = mean(M) \times VSP \quad (2)$$

Where tor : neighbor vector per row, r: is a positive value such that 1 ≤ r ≤ nbr.
VSP is vector of SP (accumulating the value of SP), and the size of this vector depends on limit value.

Table 2: Example of computing Nvector value of the first mean, where the number of neighbors(nbr) $= 8$, and the step size (SP) = 0.25.

$\boldsymbol{az}$	$\boldsymbol{t(1)}$	$\boldsymbol{t(2)}$	$\boldsymbol{t(3)}$	$\boldsymbol{t(4)}$	$\boldsymbol{t(5)}$	$\boldsymbol{t(6)}$	$\boldsymbol{t(7)}$	$\boldsymbol{t(8)}$
$VSP = 0.25$	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$Nvector(t)$	22	45	67	90	112	135	157	180

Step 5: find the index & residual array through the following steps:
Calculate and obtain the smallest positive value that is left over after a set of divisions between the current pixel value and the (Nvector) values of that row for every pixel in each row. This value should be positive. As shown in the following equation, the index of the (Nvector) value that produces the smallest remaining value is stored in a matrix called *Indx*, which is considered an intermediate matrix. This equation shows the relationship between the nearest value of the (Nvector) and the current pixel value. After this, the index (t) of the value that is closest to the current one is stored.

$$Indx(m,t) = (J(m,n)/Nvector) \quad (3)$$

Where $Indx$ It is in term array of indices for the Nvector that yields the smallest remaining value.

The value of Nvector(t) is subtracted from the value of J (m, n), and the t corresponds to the smallest positive value that results from this subtraction.
Calculate the remainder (residual) between the current pixel value J (N, N) and the selected (nearest) Nvector(t). This should be done in such a way that the following Eq. (4) is satisfied:

$$Res\ (N,N) = J(N,N)\ -\ Nvector(r) \quad (4)$$

Where Res is an array of reminder values (residual);(r) represents the times the current pixel value is more significant than Nvector. Save zero at the Indexes array cell if the current pixel value is smaller than all Nvector values.

Table 3: Example of computing index(Indx) & residual array(Res) of the first mean, where the number of neighbors(nbr) $= 8$, and the step size (SP) = 0.25 & J(m,n)=185.

$\boldsymbol{nbr}$	$\boldsymbol{t(1)}$	$\boldsymbol{t(2)}$	$\boldsymbol{t(3)}$	$\boldsymbol{t(4)}$	$\boldsymbol{t(5)}$	$\boldsymbol{t(6)}$	$\boldsymbol{t(7)}$	$\boldsymbol{t(8)}$
$VSP = 0.25$	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$Nvector(t)$	22	45	67	90	112	135	157	180
$Indx(1,t)$	8	4	2	2	1	1	1	1
$Res(1,t)$	162	140	112	95	72	50	27	5

The residual matrix will be divided into two parts: MSV & LSV, such as

$$ResMSV = round(\frac{RES}{Stp}) \quad (5)$$

$$ResLSV = RES\ -\ ResMSV\ \times\ Stp \quad (6)$$

Stp = is the value of the bit that represents the cut point. It is a highly significant parameter that divides the values into several ranges [30-150].
Step 5: Scalar uniform quantization/dequantization was used for the LSV residual in the beginning by using the quantization step (Qstep).

$$Rlsv_Q = round\ (\frac{ResLSV}{Qstep}) \quad (7)$$

$$Rlsv_{DQ} =\ ResLSV \times\ Qstep \quad (8)$$

Where $Rlsv_Q$ = quantized residual image, $Rlsv_{DQ}$ = dequantized residual image
Applied DWT of quantized residual ($Rlsv_Q$) and after two levels are obtained, a hard threshold is applied to bands Low High (LH), High Low (HL), and High High (HH) at both

levels because these bands contain small and unimportant details such as Eq. (9) [22].At the same time, the essential information is collected on band LL (Low Low).

$$Hard\ Threshold = am(X)\begin{cases} X & if\ |X| \geq T \\ 0 & if\ |X| < T \end{cases} \quad (9)$$

Where am = is the threshold function, X = is the coefficient value, and T= is the threshold value.
Encode /decode the four bands (LL, HL, LH, HH) loosely by mixing two coding techniques, Huffman and LZW.

Step 6: Apply the Minimize Matrix Size Algorithm (MMSA) to the second part of the residual (Resmsv), According to the equation below, it losslessly represents each of three data values into a single floating value by exploiting weighted random values between [0] and [1].

$$M_{x3}\ (i,j) = \sum_{t=1}^{M_s} K(t) \times M3(i, j+m) \quad (10)$$

Where M_s equals 3, K is the number of random keys, and m is the row size (0, 3, 6, 9, etc.). Increased by three; i and j are the row and column coordinates, M3 is the corresponding redundancy image, and MM3 is the compressed minimized matrix based on three data (the sum of the products of the weights and the original data values). As described in the following sub-steps:
1- Initialize three pointers of empty values (P (1) = P (2) = P(3) = 0), where each pointer corresponds to the location (position) of an array in the Limited Space Search Table (LSST).
2- All pointers refer to the first position (location) in LSST.
3- Reserve the value of the third parameter P (3) in a vector called Support Data to facilitate (speed up) the process of retrieval of data by searching for the first and second parameters only.
4- Start with the last pointer, P (3), and increase it by one at each iteration, which means using Eq. (10) until reaching the last location (position) of LSST.
5- Encode/decode the Limited Space Search Table (LSST) losslessly by combining the encoder using Huffman and LZW coding techniques.

$$Rec\_M_{x3}\ (i,j) = \sum_{t=1}^{M_s} K(t) \times LSST\big(P(t)\big) + SP(t+2) \quad (11)$$

Where Rec_M_{x3} refers to the estimated Minimize Matrix Size, LSST is Limited Space Search Table array, P refers to pointers, K refers to utilized Keys, M_s =3 represents a number of compressed values, SP refers third parameter value of input data.

8.a. Encoding

This technique generates two arrays: the first matrix (enc_ Res_m) includes the encoded data, while the second matrix (SP_ Res_m) contains the third parameter of the original data in a row. As mentioned previously, the basis of this technique is to convert all three parameters to a single parameter and store it with the third parameter to facilitate and speed up the retrieval process.
Encode/decode the (enc_ Res_m) losslessly by using the Huffman technique, while the (SP_ Res_m) is encoded by combining two coding techniques, LZW and Huffman.

8.b. Decoding

To restore the original data of MSV residual, start with LSST to find the two compressed values; the third value is known and stored in support data. Using the same keys used for the first time in the encoding process, the original data can be recovered without loss.

Step 9: To reconstruct the image's original, the sub-mean value of each row needs to be recalculated. This is because the decoding unit consists of three arrays (index, mean, and residual), depending on the parameters (nbr, SP). According to the following equations and Table 4.

$$R\_mean = Nvector(m) \times SP \times r \tag{12}$$

Where SP refers to step size ; r It represents the lowest value in the residual array.

$$J\_Reconstruct\ (m,n)\ = indx(m,n) \times SP + J\_res(m,n) \tag{13}$$

Where J_res =is the inverse splitting (LSV&MSV) array of residual array.

$$Rvmean = 90 \times 0.25 \times 8 = 180$$

$$J - Rec(1,1)\ =\ 1 \times 180\ +\ 5\ =\ 185$$

Table 4: Example to reconstruct The value of $pixel\, J(m,n)$

No.of row	Mean(m)	$Rvmean$	$Res(m,r)$	J_Rec(m,n)
1	90	180	5	185

Where$Res(m,r)$ denotes the smallest positive value in the residual matrix m,r represents the row and column, respectively, of the index array.

6. Results and Discussion

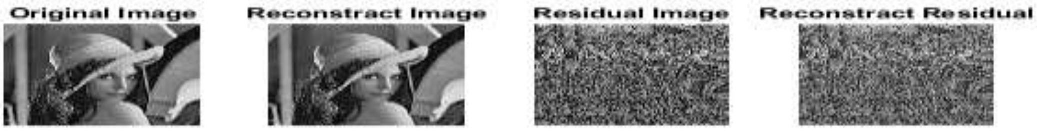
The main goal of any compression process is to obtain a small file size while maintaining quality [23]. Generally, the quality appearance or quality performance measure the loss or degradation due to exploiting the quantization process, which represents the source of loss, and PSNR is the difference in distortion between two images [24, 25]. To obtain the best results, a balance must be struck between size and quality. In [15], the obstacle was the large size of the residual. This work shows the superiority of pixel-based techniques' (adaptive/enhanced) performances in reducing size and improving PSNR, which improved the inherited problems of traditional pixel-based problems with large residuals and the need for overhead information (index, mean).

$$MSE\big(J,\hat{J}\ \big) = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\,\hat{J}(i,j) - J(i,j)]^2 \tag{14}$$

Where J original image; $\hat{J}$ is compressed image, N×N is a square image of N rows and N columns

$$PSNR = \big(J,\hat{J}\big) = 10\, log\, 10 \left(\frac{(255)^2}{MSE}\right) \tag{15}$$

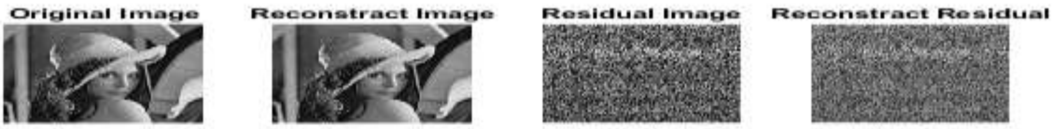
In this work, the well-known grayscale images (8 bits/pixel) with a size of (256 × 256) were used. The performance of this system is affected by the parameters used (nbr and SP) and quantization parameters (Qstep, fac, and stp). And it was found that the value of fac had an effect on the size; the greater the value of fac, the smaller the sample, and in return, the value of the PSNR increases. A sample of this method used on a standard image of a Lena gray image with various control parameters is shown in Figure 2. It is obvious how the number of neighbors and step size affected the residual image and how the quantization value and thresholding value affected the quality of the images. As the number of neighbors (nbr), step size (SP), and Qstep increase, the size residual (size_Res) decreases at the expense of image quality, and vice versa, where size_Res increases while preserving image quality from deterioration.



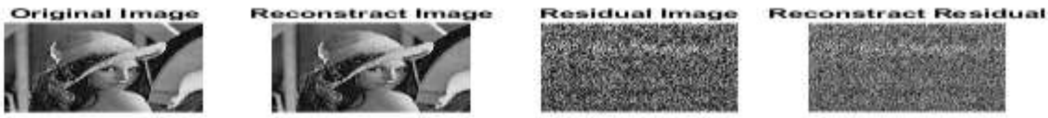
size_Res=6484, PSNR=48.6219, nbr= 10, SP= 0.25, Qstep=2, stp= 10, fac=0.7

size_Res=6128, PSNR=48.6053, nbr= 20, SP = 0.125, Qstep=2, stp= 10, fac=0.7

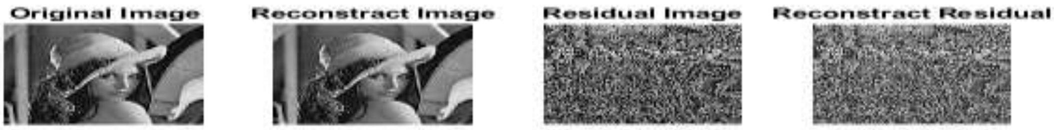
size_Res=5656, PSNR= 45.6320, nbr= 10, SP= 0. 25, Qstep=4, stp= 10, fac=0.7



size_Res=5346, PSNR=45.7576, nbr= 20, SP = 0. 125, Qstep=4, stp= 10, fac=0.7



size_Res=5624, PSNR= 45.5044, nbr= 10, SP = 0. 25, Qstep=4, stp= 10, fac=0.9



size_Res=5346, PSNR= 45.7576, nbr= 20, SP = 0. 125, Qstep=4, stp= 10, fac=0.9

Figure 2: Enforcement of proposed method on gray image Lena

The tables below show the results of the previous work in terms of residual size, PSNR, and the values of the mean (size_mean) and index (Size_index) which are obtained through Eqs. 1 and 3, respectively, in both cases when the number of neighbors is 10 and the other case when it is 20, and the value of the increase between these neighbors is 0.25 and 0.125, respectively.

Table 5: Enforcement of Traditional Technique on Lena image (1)

	State:1 Size_mean= 78 byte SP=0.25				**No of neighbors=10 Size_index=1276 byte**		**State: 2 Size_mean=78 byte SP=0.125**		**No of neighbors =20 Size_index=2234 byte**	
	a	b	q=20		q=30		q=20		q=30	
			Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR
1	0.8	0.6	10132	40.486	7897	37.182	6416	40.438	3912	38.626
2	0.9	0.8	9690	40.417	7453	37.117	5986	40.373	3478	38.537
3	1	1	9319	40.329	7124	37.018	5621	40.283	3194	38.406
4	1.1	1.2	9026	40.216	6808	36.893	5362	40.169	2912	38.264
5	1.2	1.4	8754	40.045	6567	36.720	5127	40.015	2696	38.105
6	1.3	1.6	8505	39.812	6350	36.508	4921	39.828	2496	37.982

Table 6: Enforcement of Traditional Technique on Cameraman image (2)

	State:1						State: 2			
	Size_mean= 110 byte SP=0.25				No of neighbors=10 Size_index=950 byte		Size_mean=110 byte SP=0.125 byte		No of neighbors =20 Size_index=1954	
	a	b	q=20		q=30		q=20		q=30	
			Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR
1	0.8	0.6	8467	42.6084	6712	39.5903	6460	42.4718	4540	40.1563
2	0.9	0.8	8080	42.5294	6347	9.5123	6067	42.3903	4141	40.0551
3	1	1	7789	42.4164	6082	39.4027	5747	42.2807	3874	39.9161
4	1.1	1.2	7553	42.2871	5835	39.2549	5506	42.1376	3613	39.7514
5	1.2	1.4	7331	42.1033	5638	39.0554	5277	41.9447	3434	39.5638
6	1.4	1.7	7130	41.8760	5451	38.8025	5078	41.7204	3245	39.3793

Table 7: Enforcement of Traditional method on Rose image (3)

	State:1						State: 2			
	Size_mean= 92 byte SP=0.25				No of neighbors=10 Size_index=1000 byte		Size_mean=92 byte SP=0.125 byte		No of neighbors =20 Size_index=1848	
	a	b	q=20		q=30		q=20		q=30	
			Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR	Size_Res	PSNR
1	0.8	0.6	9483	41.8743	7572	39.0469	6790	41.7309	4698	39.2969
2	0.9	0.8	9060	41.7883	7140	38.9573	6381	41.6459	4266	39.2022
3	1	1	8704	41.6740	6827	38.8236	6019	41.5292	3965	39.0706
4	1.1	1.2	8430	41.5276	6531	38.6665	5757	41.3938	3701	38.9235
5	1.2	1.4	8176	41.3210	6303	38.4493	5510	41.2071	3504	38.7406
6	1.3	1.6	7940	41.0491	6088	38.1646	5306	40.9857	3316	38.5658

In the tables below, three images were tested in the proposed system, and each image has two cases according to the number of neighbors (nbr) and the step size (SP); the best value for quantization is 4, and the best values that can be obtained for thresholding values when the value of fac >= 0.7 and the value of stp = 10, by the equations. (1 and 3). The size mean (size_mean) and size index (Size_index) values are calculated, respectively.

Table 8: Enforcement of proposed method on Lena image (1)

	State 1:			size_mean=78 byte	State 2:	size_mean=110 byte
	no of neighbor=10			SP=0.25	no of neighbor=20	SP=0.125
	Stp =10			size_index =1276 byte	Stp =10	size_index =2234 byte
	fac	**Qstep**	**Size_Res**	**PSNR**	**Size_Res**	**PSNR**
1	0.1	2	7988	51.2106	7624	51.313
2	0.1	4	6470	47.0359	6160	47.2215
3	0.1	8	5840	39.9896	5558	40.3146
4	0.1	10	5132	39.1265	4856	39.4516
5	0.1	16	4098	39.1265	3854	39.4516
6	0.2	2	7572	51.0798	7230	51.1706
7	0.2	4	6470	47.0359	6160	47.2215
8	0.2	8	5840	39.9896	5558	40.3146
9	0.2	10	5132	39.1265	4856	39.4516
10	0.2	16	4098	39.1265	3854	39.4516

11	0.2	2	7572	51.0798	7230	51.1706
12	0.5	2	6834	49.7578	6474	49.7923
13	0.5	4	6062	46.6769	5748	46.8508
14	0.5	8	5368	39.9402	5438	40.302
15	0.5	10	5132	39.1265	4856	39.4516
16	0.5	16	4098	39.1265	3854	39.4516
17	0.7	2	6484	48.6219	6128	48.6053
18	0.7	4	5656	45.632	5346	45.7576
19	0.7	8	5368	39.9402	5072	40.2547
20	0.7	16	4098	39.1265	3854	39.4516

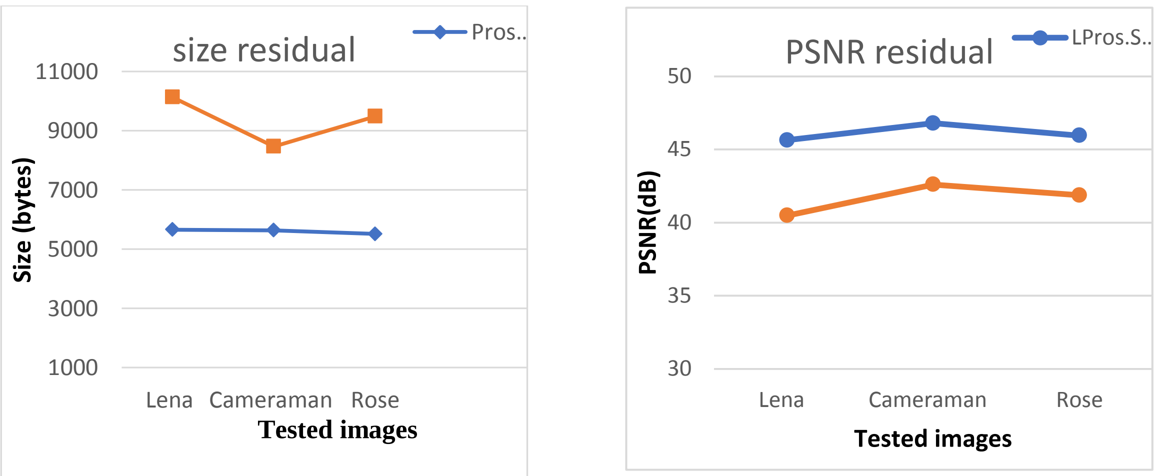
Table 9: Enforcement of proposed method on Cameraman image (2)

State 1:				**size_mean=110 byte**	**State 2:**	**size_mean=110 byte**
	no of neighbor=10			**SP=0.25**	**no of neighbor=20**	**SP=0.125**
Stp =10				**size_index =950 byte**	**Stp =10**	**size_index =1954 byte**
	Fac	**Qstep**	**Size_Res**	**PSNR**	**Size_Res**	**PSNR**
1	0.1	2	7568	51.2542	7240	51.2819
2	0.1	4	6138	47.1118	5868	47.0647
3	0.1	8	5602	39.7648	5272	40.2623
4	0.1	10	4924	38.8123	4622	39.4219
5	0.1	16	3970	38.8123	3738	39.4219
6	0.2	2	7184	51.2046	7024	51.2286
7	0.2	4	6138	47.1118	5868	47.0647
8	0.2	8	5602	39.7648	5272	40.2623
9	0.2	10	4924	38.8123	4622	39.4219
10	0.2	16	3970	38.8123	3738	39.4219
11	0.5	2	6652	50.7612	6332	50.7749
12	0.5	4	5704	46.9701	5432	46.9318
13	0.5	8	5462	39.7678	5196	40.2487
14	0.5	10	4924	38.8123	4622	39.4219
15	0.5	16	3970	38.8123	3738	39.4219
16	0.7	2	6236	49.6474	5954	49.7972
17	0.7	4	5638	46.7977	5370	46.7828
18	0.7	8	5168	39.8184	4822	40.3034
19	0.7	10	4924	38.8123	4622	39.4219
20	0.7	16	3970	38.8123	3738	39.4219

Table 10: Enforcement of proposed method on Rose image (3)

			State 1: size_mean=92 byte, no of neighbor=10, SP=0.25, Stp =10, size_index =1000 byte		State 2: size_mean=92 byte, no of neighbor=20, SP=0.125, Stp =10, size_index =1848 byte	
	Fac	**Qstep**	**Size_Res**	**PSNR**	**Size_Res**	**PSNR**
1	0.1	2	7764	51.4632	7546	51.5524
2	0.1	4	6292	47.2840	6038	47.3950
3	0.1	8	5708	40.1733	5428	40.1773
4	0.1	10	5024	39.2744	4724	39.2693
5	0.1	16	4020	39.2744	3730	39.2693
6	0.2	2	7322	51.3411	7094	51.4216
7	0.2	4	6292	47.2840	6038	47.3950
8	0.2	8	5708	40.1733	5428	40.1773
9	0.2	10	5024	39.2744	4724	39.2693
10	0.2	16	4020	39.2744	3730	39.2693
11	0.5	2	6698	50.3495	6462	50.3478
12	0.5	4	5884	46.9937	5618	47.0624
13	0.5	8	5430	40.1637	5122	40.1662
14	0.5	10	5024	39.2744	4724	39.2693
15	0.5	16	4020	39.2744	3730	39.2693
16	0.7	2	6268	48.9679	6010	48.9267
17	0.7	4	5512	45.9552	5244	45.9672
18	0.7	8	5236	40.1600	4966	40.1616
19	0.7	10	5024	39.2744	4724	39.2693
20	0.7	16	4020	39.2744	3730	39.2693

The Figure 3 below shows the efficiency of the proposed system compared to the traditional system in terms of reducing size and obtaining higher quality.



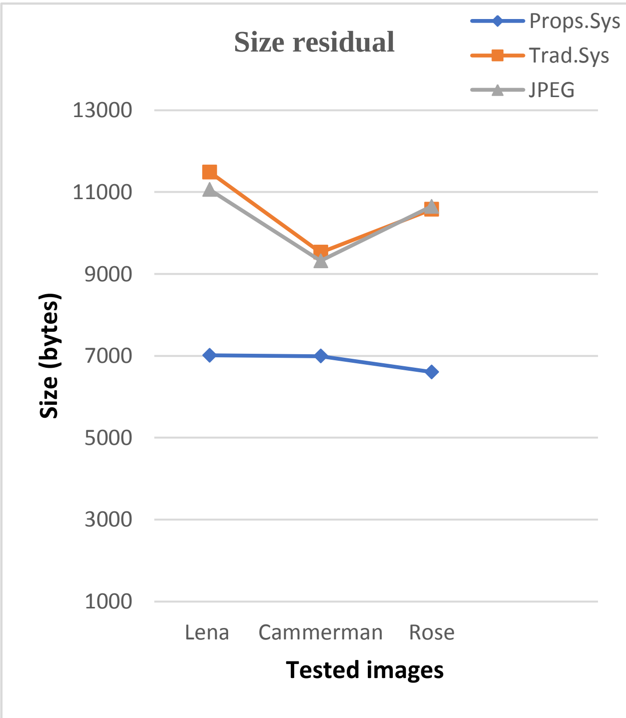
(a) **(b)**

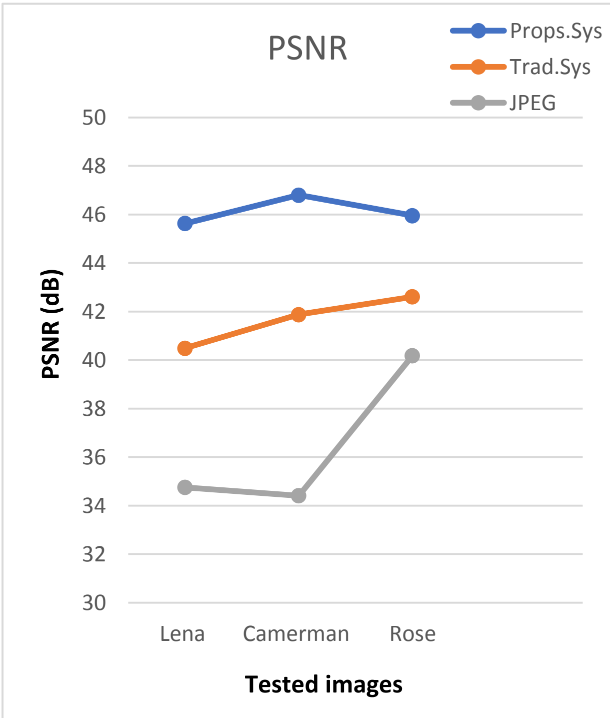
Figure 3: The comparison performance between the traditional pixel-based technique and the proposed system in terms of a) size residual and b) PSNR for tested images

Finally, a performance comparison between the traditional system and the known standard JPEG with Qstep = 4 adopted in the proposed system showed superiority in size residual and PSNR, as shown in Figure 4.

Table 10: Comparison of performance with Traditional system [15] using Qstep=20 and standard JPEG

Tested image	Traditional system				Proposed system				JPEG	
	nbr=10 SP=0.25 q=20 a =1.3 b=1.6		nbr=20 SP=0.125 q=20 a =1.3 b=1.6		nbr=10 SP=0.25 Qstep=4 fac=0.7		nbr=20 SP=0.125 Qstep =4 fac=0.7			
	Size in byte s	**PSNR**	**Size in byte**	**PSNR**	**Size in byte**	**PSNR**	**Size in byte**	**PSNR**	**Size in byte**	**PSNR**
Lena	11486	40.4866	8728	40.4387	7010	45.6320	7658	45.7576	11059	34.7484
Cameraman	9527	41.8743	8524	41.7309	6992	46.7977	6724	46.7828	9318	34.4102
Rose	10575	42.6084	8730	41.7309	6604	45.9552	7184	45.9672	10649	40.1794





(a) **(b)**

Figure 4: The comparison performance between the standard JPEG, the proposed system, and the traditional system [15] in terms of a) Size system and b) PSNR for tested images.

7- Conclusion

The proposed adaptive pixel-based coding techniques of hybrid base that mix between lossless and lossy schemes for deterministic and probabilistic parts, respectively, also imply the incorporation of MMSA in C321 base for coding the residual part efficiently, which leads to high packing compared to the traditional techniques and the standard JPEG. The work may extend to involve color and/or non-square images, along with selective DPCM and non-uniform quantization.

Reference

[1] G. K. AL-Khafaji, M. H. Rasheed, M. M. Siddeq, and M. A. Rodrigues, "Adaptive Polynomial Coding of Multi-base Hybrid Compression, " *International Journal of Engineering*, vol. 36, no. 2, pp. 236–252, 2023, doi: 10.5829/ije.2023.36.02b.05.

[2] A. S. Abd-Alzhra and M. S. H. Al- Tamimi, "Image Compression Using Deep Learning: Methods and Techniques," *Iraqi Journal of Science*, vol. 63, no. 3, pp. 1299–1312, 2022, doi: 10.24996/ijs.2022.63.3.34.

[3] N. S. Mahdi and G. K. AL-Khafaji, "Image Compression using Polynomial Coding Techniques : A review, " *Journal of Al-Qadisiyah for Computer Science and Mathematics,* vol. 14, no. 2, pp. 70–81, 2022, doi:10.29304/jqcm.2022.14.2.968 1.

[4] G. K. Al-Khafaji and L. E. Gorrge, "Grey-level image compression using 1-d polynomial and hybrid encoding techniques, " *Journal of Engineering Science and Technology.*, vol. 16, no. 6, pp. 4707–4728, 2021.

[5] Z. J. Ahmed, L. E. George, and Z. S. Abduljabbar, "Fractal image compression using block indexing technique: A review, " *Iraqi Journal of Science,* vol. 61, no. 7, pp. 1798–1810, 2020, doi: 10.24996/ijs.2020.61.7.29.

[6] A. K. Pawar and P. J. Mishra, "Image Compression Techniques for Medical Science Image "*International Journal of Innovative Research in Technology and Management,* " vol. 3404, no. 1, pp. 63–68, 2021.

[7] Z. H. Abed and G. K. Al-Khafaji, "Pixel Based Techniques for Gray Image Compression : A review, " *Journal of Al-Qadisiyah for Computer Science and Mathematics,* vol. 14, no. 2, pp. 59–70, 2022, doi.org/10.29304/jqcm.2022.14.2.967.

[8] G. Al-Khafaji, "Linear polynomial coding with midtread adaptive quantizer, " *Iraqi Journal of Science,* vol. 59, no. 1C, pp. 585–590, 2018, doi: 10.24996/IJS.2018.59.1C.15.

[9] R. I. Yousif and N. H. Salman, "Image compression based on arithmetic coding algorithm, " *Iraqi Journal of Science* , vol. 62, no. 1, pp. 329–334, 2021, doi: 10.24996/ijs.2021.62.1.31.

[10] S. G. Mohammed, S. S. Abdul-Jabbar, and F. G. Mohammed, "Art Image Compression Based on Lossless LZW Hashing Ciphering Algorithm, " *3rd International Conference in Physical Science and Advanced Materials.*, vol. 2114, no. 1, 2021, doi: 10.1088/1742-6596/2114/1/012080.

[11] A. J. Hussain, A. Al-Fayadh, and N. Radi, "*Image compression techniques: A survey in lossless and lossy algorithms*, "vol. 300. 2018. doi: 10.1016/j.neucom.2018.02.094.

[12] H. Rhee, Y. Il Jang, S. Kim, and N. I. Cho, "Lossless Image Compression by Joint Prediction of Pixel and Context Using Duplex Neural Networks, " *IEEE Access*, vol. 9, pp. 86632–86645, 2021, doi: 10.1109/ACCESS.2021.3088936.

[13] N. H. Salman and S. Rafea, "THE ARITHMETIC CODING AND HYBRID DISCRETE WAVELET AND COSINE TRANSFORM APPROACHES IN IMAGE COMPRESSION, " *JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY* ,vol. 55, no. 1, pp. 1–9, 2020. doi:10.35741/issn.0258-2724.55.1.6

[14] Y. A. Salih, A. Ali Mohammed, and L. Edwar George, "Improved Image Compression Scheme Using Hybrid Encoding Algorithm, " *Kurdistan Journal of Applied Research,* vol. 4, no. 2, pp. 90–101, 2019, doi: 10.24017/science.2019.2.9.

[15] A. A. Hussain and G. K. AL-Khafaji "A Pixel Based Method for Image Compression, " *Tikrit Journal of Pure Science* , pp. 1813–1662,2021, doi:10.25130/tjps.26.2021.018.

[16] M. M. Siddeq and M. A. Rodrigues, "A novel high-frequency encoding algorithm for image compression, "*EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, 2017, doi: 10.1186/s13634-017-0461-4.

[17] S. S. Al-hadithy, G. K. Al-Khafaji, and M. M. Siddeq, "Adaptive 1-D Polynomial Coding of C621 Base for Image Compression, "*Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 13, pp. 5720–5731, 2021.

[18] B. A. Sultan and L. E. George, "Color image compression based on spatial and magnitude signal decomposition, " *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4069–4081, 2021, doi: 10.11591/ijece.v11i5.pp4069-4081.

[19] E. L. L. Cabral, G. Sabundjian, and N. Conti, "Pixel-Position-Based Lossless Image Compression Algorithm," *International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET)* , vol. 4863, pp. 21–30, 2019.

[20] S. Anitha, V. L. Praba, and R. S. Rajesh, "Image Compression Based On Octagon Based Intra Prediction, "*Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 3, pp. 6144–6151, 2021.

[21] M. H. Rasheed, O. M. Salih, M. M. Siddeq, and M. A. Rodrigues, "Image compression based on 2D Discrete Fourier Transform and matrix minimization algorithm, " *Array*, vol. 6, no. February, 2020, doi: 10.1016/j.array.2020.100024.

[22] K. HAGIWARA, "Bridging between Soft and Hard Thresholding by Scaling, "*IEICE Transactions on Information and Systems*,vol.E105.D, no. 9, pp.1529–1536. 2022, doi: 10.1587/transinf.2021EDP7223.

[23] A. S. Ali, M. S. H. Al-Tamimi, and A. A. Abbood, "Secure image steganography through multilevel security, " *International Journal of Innovation, Creativity and Change.*, vol. 11, no. 1, pp. 80–103, 2020.

[24] R. J. Toama, N. H. M. Ali, "A Secure Cipher for the Gray Images Based on the Shamir Secret Sharing Scheme With Discrete Wavelet Haar Transform," *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*, vol. 15, no. 6, pp. 334–351, 2020. doi:0.26782/jmcms.2020.06.00025.

[25] A. Mandegari, E. Akhtarkavan, and B. Majidi, "Secure Medical Image Communication Using Fragile Data Hiding based on Discrete Wavelet Transform and A5 Lattice Vector Quantization ," *IEEE Access*,vol.11, pp. 9701–6715, 2023, doi: 10.1109/access.2023.3238575.