# Lossy Compression using Adaptive Polynomial Image Encoding

Shaimaa OTHMAN[1, 2], Amr MOHAMED[1], Abdelatief ABOUALI[2], Zaki NOSSAIR[1]
[1]*Department of Electronics, Communications, and Computers, Faculty of Engineering, Helwan University, Cairo, 11795, Egypt*
[2]*Institute of Engineering, El-Shorouk Academy, Cairo, 11837, Egypt*
*shaimaaosman20111@hotmail.com*

***Abstract*—In this paper, an efficient lossy compression approach using adaptive-block polynomial curve-fitting encoding is proposed. The main idea of polynomial curve fitting is to reduce the number of data elements in an image block to a few coefficients. The proposed approach consists of two processes: encoding and decoding. In the encoding process, the coefficient matrix is created by representing each block of the image with a first- or second-order two-dimensional polynomial. The encoded block size of the image is variable. The polynomial order and the encoded block size are determined dynamically depending on the value of a threshold. A prefix code of two bits is used to differentiate the encoding states. Uniform quantization is applied to the coefficient matrix to store these coefficients effectively. In the decoding process, the reconstructed (decompressed) image is built from the quantized coefficient matrix. The fitting variables are two-dimensional (x, y). The encoding and decoding processes require a single image scan without the need to transfer the matrix to another domain. Experimentally, a high compression ratio is achieved at an acceptable quality for both gray and color images. The results are comparable to those of most recent studies.**

***Index Terms*—adaptive, compression, decoding, encoding, polynomial.**

## I. INTRODUCTION

In recent decades, image processing and its applications have been a focal point of modern research due to the increasing demand for these applications. These applications are image enhancement [1], image restoration [2], image segmentation [3], image representation [4], image detection [5], image recognition [6], image steganography [7], content-based image retrieval [8], and image encoding [9].

An image encoding technique is a process that converts a set of characters such as numbers, letters, symbols, and punctuation into a particular format to better meet the transmission and storage requirements. The encoded media files are put into an efficient format called a compressed format. The encoded media files should be similar to the original uncompressed files, but it is preferable to have small file sizes. The decoding process converts the encoded media file back into the original file. Encoding and decoding processes are used in many applications, such as networking, data communication, and storage, to facilitate the transmission process. Efficient encoding algorithms have a high compression ratio, preserve information, and match modern computer architectures. Both encoding and decoding algorithms have a low complexity order [10], [11].

The main classification perspectives of image encoding algorithms are the coding length, application domain, and information loss. Regarding coding length, there are two basic types: fixed and variable lengths. In fixed-length encoding, the length of all coding units is the same [12]. However, in variable-length encoding, the coding units can be of different sizes [13], [14]. Fixed-length encoding algorithms include run-length encoding [15], [16], and variable-length encoding algorithms include Lempel–Ziv coding [17], Huffman coding [18], [19], and arithmetic coding [20], [21].

From the application domain perspective, the encoding process can be performed in the spatial domain, or it can require transformation to another nonspatial domain. Spatial domain encoding saves transformation costs [22]. However, in the nonspatial (frequency) domain, a transform is applied with the aim of distributing image energy in a form that is more suitable for subsequent effective encoding steps. Examples of such transforms are the Fourier transform (FT) [23], wavelet transform (WT) [24], and discrete cosine transform (DCT) [25], [26].

From the information-loss perspective, there is lossless compression (reversible/exact) and lossy compression (irreversible) [11]. In reversible (lossless) images, the decoded image is identical to the original image [27]. Lossless schemes include Lempel–Ziv coding [17], Huffman coding [18], [19], and arithmetic coding [20], [21]. In lossy schemes, a controlled loss is allowed to maximize the compression without affecting the efficiency of the media file [28]. Lossy encoding schemes include JPEG [29], fractal [30], wavelet [31], and perceptual coding [32].

Vector quantization (VQ) is an effective method that saves storage and bandwidth for image, speech, and other sensor data acquisition. The primary function of the vector quantizer is to map a sequence of discrete or continuous vectors into a more compact digital sequence. The quantizer generates a codebook that is known by the encoder and decoder. The encoded data are written using codebook indices [31], [33].

The encoding algorithm requires metrics to measure its performance. Here, we focus on four basic encoding metrics: the root-mean-square error (RMSE), peak signal-to-noise ratio (PSNR), compression ratio (CR), and similarity structure index metric (SSIM) [34-36]. Assume we have an $m x n$ image, the uncompressed image $f(i,j)$, and the reconstructed (decompressed) image g(i,j), where $i = 1, 2, \dots m$ and $j=1, 2, \dots n$. The RMSE is the cumulative RMSE between the uncompressed image and the reconstructed (decompressed) image, which is defined as [35]

Digital Object Identifier 10.4316/AECE.2021.01010

$$RMSE=\sqrt{\frac{1}{mxn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|f\left(i,j\right)-g\left(i,j\right)\right|^{2}} \quad (1)$$

So, the RMSE is a measure for the average difference between the pixel values of the original image and the pixel values of the reconstructed image. Therefore, the RMSE units are the same as image matrix values, intensity.

The PSNR is the ratio between the maximum power of the signal of the original image and the noise that occurs in the compression process, which is defined as [34]

$$PSNR=20log_{10}\left[\frac{2^{b}-1}{RMSE}\right] \quad (2)$$

where, $b$ is the pixel depth in bits. Better compression is achieved when the RMSE is low and the PSNR is high. The CR or the reduction percent is the ratio between the compressed and uncompressed image sizes, which is defined as [35]

$$CR=\frac{Compressedimagesize}{Uncompressedimagesize}\% \quad (3)$$

The SSIM measures the range of structural variation of the reconstructed image over the original image. Its value ranges between 0.0 and 1.0; a higher value means a smaller structural variation and vice versa. The SSIM is defined as [36]

$$SSIM\left(x,y\right)=\frac{\left(2\mu_{x}\mu_{y}+c_{1}\right)\left(2\sigma_{xy}+c_{2}\right)}{\left(\mu_{x}^{2}+\mu_{y}^{2}+c_{1}\right)\left(\sigma_{x}^{2}+\sigma_{y}^{2}+c_{2}\right)} \quad (4)$$

where, $\mu_x$ and $\mu_y$ are the averages of $x$ and $y$, respectively; $\sigma_x$ and $\sigma_y$ are the variances of $x$ and $y$, respectively; $\sigma_{xy}$ is the covariance of $x$ and $y$; $c_1=(k_1L)^2$ and $c_2=(k_2L)^2$ are two variables that stabilize division with weak denominators; $L$ is the dynamic range of the pixel values; and $k_1=0.01$ and $k_2=0.03$ by default.

## II. Related Work

The importance of image encoding is what has attracted many researchers to this field. The research efforts in this domain started as early as the beginning of the digital era. This section will go through some of the encoding and decoding efforts that are related to our study.

R. J. Al-Bahadili proposed a technique to compress an image using two different polynomial models depending on the variance of the block pixels after dividing the image into fixed-size blocks and then applied multiple levels of scalar quantization and Huffman encoding to obtain the reconstructed image [35].

S. Ameer presented an image compression algorithm that depends on plane fitting. The algorithm starts by dividing the image into nonoverlapping blocks and applying plane fitting to each block, which represents each block with three or four coefficients; then, a quantization method and coding schemes are applied [37].

S. Sadanandan introduced a lossy compression method to eliminate redundancy by using skipline encoding and curve-fitting-based encoding [38].

S. Sajikumar proposed a technique to compress an image by using a Chebyshev polynomial surface fit. The encoding is carried out by dividing the image into fixed blocks and applying the Chebyshev polynomial that represents each block in the image with three or four coefficients [39].

E. Erdal introduced a Huffman encoding algorithm that is used to decrease the number of bits, represented by long-bit codewords. The algorithm begins by applying the Huffman encoding algorithm to the data. The characters that are represented by short-bit sequences in the Huffman encoding algorithm are ignored. Then, if consecutive symbols are on the same leaf, flag bits are added. A flag bit “0” is added if the next character is not on the same leaf; otherwise, the flag bit “1” is added between the characters [40].

W. Khalaf presented a technique to compress an image using a curve-fitting model derived from a symmetric function (hyperbolic tangent). The image is divided into fixed 4x4 blocks. The symmetric function uses only three coefficients to fit each block [41].

Y. Zhang proposed a depth data compression scheme for depth images. A platelet-based coding method that depends on quadtree decomposition, the Lagrangian optimization function, and the quadratic curve-fitting algorithm was used. First, the depth map is divided into blocks with one or two homogeneous regions using quadtree decomposition. Second, the distortion is minimized by Lagrangian optimization. Finally, four quadratic curve-fitting models are built for the contours of the objects to retain the position of the edges more precisely after compression [42].

The rest of this paper is organized as follows: Section III explains the proposed encoding and decoding processes, Section IV describes the experimental results, and, finally, the study conclusion and future work are presented in Section V.

## III. Proposed Technique

The proposed technique is a block-based variable-length lossy encoding scheme. The encoding and decoding techniques are applied to a two-dimensional image matrix. The encoding method is applied directly to one or three consecutive two-dimensional matrices for gray and color images, respectively. The encoding process consists of two phases: polyfitting and quantization.

### A. The Encoding Process

The polyfitting phase uses two block sizes, ($MxM$) and ($M/2xM/2$), where $M$ is divisible by 2. To control the loss, a predetermined RMSE threshold, TH, is set. Two types of polynomials are employed, first- and second-order polynomials, as in Eqs. (5) and (6), respectively:

$$p\left(x,y\right)=p_{0}+p_{1}x+p_{2}y \quad (5)$$

$$p\left(x,y\right)=p_{0}+p_{1}x+p_{2}y+p_{3}x^{2}+p_{4}y^{2}+p_{5}xy \quad (6)$$

where, $p_0, p_1, p_2, p_3, p_4$ and $p_5$ are the polynomial coefficients.

Therefore, either three or six coefficients are generated in a block. The details of the polyfitting phase algorithm are shown in Algorithm 1.

| Algorithm 1: Polyfitting phase |
|---|
| 1: Divide the image into blocks of size (*MxM*).<br>2: Encode an (*MxM*) block using first-order polynomial fitting.<br>3: Calculate the RMSE.<br>4: If (RMSE<=TH)<br>-Write (Encoded-Block).<br>Else<br>{<br>-Encode the (*MxM*) block using second-order polynomial fitting.<br>-Calculate the RMSE.<br>}<br>5: If (RMSE <= TH)<br>-Write (Encoded-Block).<br>Else<br>-Divide each block into 4 ((*M/2xM/2* blocks)).<br>5.1: For (Block-Index=0; Block-Index<4; Block-Index++)<br>{<br>-Encode an (*M/2xM/2*) block of the Block-Index using first-order polynomial fitting.<br>-Calculate the RMSE.<br>5.1.1: If (RMSE<=TH)<br>-Write (Encoded-Block).<br>Else<br>{<br>-Encode the (*M/2xM/2*) block of the Block-Index using second-order polynomial fitting.<br>-Write (Encoded-Block).<br>}<br>}<br>6: If (end of the image)<br>-Start the quantization phase.<br>Else<br>{<br>-Read the next (*MxM*) block.<br>-Repeat steps 2 to 6 of the algorithm.<br>} |

TABLE I. THE BLOCK CODE (BC)

| Block code | Meaning | Number of coefficients |
|---|---|---|
| 00 | *MxM* block size and first order. | 3 |
| 01 | *MxM* block size and second order. | 6 |
| 10 | *M/2xM/2* block size and first order. | 3 |
| 11 | *M/2xM/2* block size and second order. | 6 |

As a result of the polyfitting phase, a matrix of coefficients is formed, and each block is represented as a code consisting of two bits that are added before each encoded block. These two-bit codes specify both the block size ((*MxM*) or (*M/2xM/2*)) and the encoding polynomial degree (first or second order). Table I specifies these two-bit meanings. Consequently, these two bits implicitly determine the number of coefficients that represent each block, as indicated in Table I.

The quantization phase uses the simple uniform quantization process, which represents each polynomial coefficient using specific bits. The quantization process starts by determining the quantization levels that control the PSNR and CR.

The details of the quantization phase algorithm are shown in Algorithm 2.

| Algorithm 2: Quantization phase |
|---|
| 1: Let $X_{min}$ and $X_{max}$ be the minimum and maximum values of the coefficient matrix, respectively, and let *n* bits be used per coefficient.<br>2: Determine the quantization levels, *L*:<br>$L=2^n$ . (7)<br>3: Determine the quantization interval, *delta*:<br>$delta=\frac{(X_{max}-X_{min})}{L}$ (8)<br>4: For each coefficient value, $C_{value}$, the quantized value, $R_{value}$, is<br>$q_{value}=integerround\left(\frac{C_{value}\text{-}X_{min}}{L}\right)$ (9)<br>The values of $X_{min}$ and $X_{max}$ are kept in the file header of each 2D image matrix. |

## B. *The Decoding Process*

The decoding process starts by reading the image header data, which determines the image resolution and the type of image (gray or color). The encoding block position (Block-Position) within the compressed file determines the starting position (x, y) of the block. For a gray image, one image matrix is produced, and for a color image, three image matrices are produced. For each quantized value, $q_{value}$, the decoded (recovered) value $R_{value}$ is

$$R_{value}=X_{min}+q_{value}*delta \quad (10)$$

The details of the decoding algorithm are shown in Algorithm 3.

| Algorithm 3: Decoding algorithm |
|---|
| 1: Read the matrix header data to obtain $X_{min}$, $X_{\max}$, the basic encoding block size *MxM*, and the number of quantization bits $n$ .<br>2: Initialize the image matrix.<br>3: Set Block-Position to (0, 0).<br>4: While (Not end of image)<br>5: Read (Block Code (BC)).<br>6: Case: BC is 00<br>-Read the three quantized coefficient values.<br>-Calculate the recovered values of the three quantized coefficients using Eq. (10).<br>-Recover the (*MxM*) Pixel Block using the first-order polynomial fitting in Eq. (5) and place it at Block-Position. |

-Update Block-Position.
7: Case: BC is 01
-Read the six quantized coefficient values.
-Calculate the recovered values of the six quantized coefficients using Eq. (10).
-Recover the (*MxM*) Pixel Block using the second-order polynomial fitting in Eq. (6) and place it at Block-Position.
-Update Block-Position.
8: Case: BC is 10
-Read the three quantized coefficient values.
-Calculate the recovered values of the three quantized coefficients using Eq. (10).
-Recover the (*M/2xM/2*) Pixel Block using the first-order polynomial fitting in Eq. (5) and place it at Block-Position.
-Update Block-Position.
9: Case: BC is 11
-Read the six quantized coefficient values.
-Calculate the recovered values of the six quantized coefficients using Eq. (10).
-Recover the (*M/2xM/2*) Pixel Block using the second-order polynomial fitting in Eq. (6) and place it at Block-Position.
-Update Block-Position.

## IV. Experimental Results

Experimentally, the used images in the proposed technique are Lena, Cameraman, Rice, and Pepper of size 256x256 and Kahramana of size 256x128 [35], [39]. To evaluate the proposed algorithm and establish a good comparison with others, the following metrics are used: RMSE, PSNR, CR, and SSIM. The proposed algorithm is tested using both gray and colored images. In our experiment, we used the MATLAB (R2018a) program on an Intel i7-core 1.50 GHz, Windows 10, 64-bit OS with 8.00 GB RAM. The experimentation includes the effect of changing the block sizes, the effect of changing the quantization levels, a gray image experiment with a varying threshold, a color image experiment with a varying threshold, and finally a comparative study with recent publications.

### A. *The Effect of Changing the Block Sizes*

The algorithm requires the specification of block sizes and quantization levels, which are considered the main parameters that affect the overall performance. Therefore, here, we study the effects of using different block sizes (4x4 and 8x8), (8x8 and 16x16), and (16x16 and 32x32) on both the CR and the quality of the image. This experiment applies to the gray images Lena, Pepper, and Cameraman, while the quantization bits are set to 10 bits, which correspond to 1024 levels, and the threshold is set to 5. Figure 1 presents the algorithm performance for varying block sizes, and Figure 2 shows both the decompressed images with different block sizes and the original images. Figures 1 and 2 show that as the block size increases, the values of the quality metrics decrease, and, at the same time, the compression is enhanced; as the block size decreases, the reverse occurs. For example, in the Lena image, at block sizes of 16x16 and 32x32, the PSNR is 22.23 dB, and the CR is 2.89%, while, at block sizes of 4x4 and 8x8, the PSNR is 31.31 dB, and the CR is 34.77%. Additionally, the block sizes 8x8 and 16x16 seem to preserve high quality (25.91 dB) with significant compression (10.33%). Therefore, we use block sizes of 8x8 and 16x16 in our experiments.

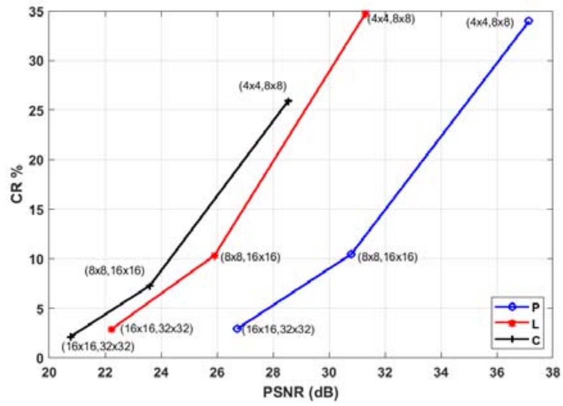



Figure 1. Performance for varying block sizes on Pepper, P; Lena, L; and Cameraman, C.


Figure 2. (a) Original image, (b) decompressed image of block sizes 4x4 and 8x8, (c) decompressed image of block sizes 8x8 and 16x16, and (d) decompressed image of block sizes 16x16 and 32x32.

### B. *The Effect of Changing the Quantization Levels*

In this section, we study the effects of each quantization level on the same set of images that are used in the previous section. The quantization bits vary from 4 to 14 bits, which correspond to quantization levels from 16 to 16384, while the block size is set to 8x8 and 16x16 with the same threshold, 5. Figure 3 presents the performance for varying quantization levels, and Figure 4 shows the decompressed images at varying quantization levels. Figures 3 and 4 indicate that as the quantization levels decrease, the decompressed images suffer from blocking and pixilation (a low PSNR), while the CR improves. Additionally, as the quantization levels increase, some saturation is seen at 8 bits, and apparent full saturation is reached at 10 bits. From the previous two experiments, we conclude that the performance is enhanced when we use block sizes 8x8 and 16x16 at a quantization level equal to 1024. Therefore, we use these parameters in our experiments on gray and color images.

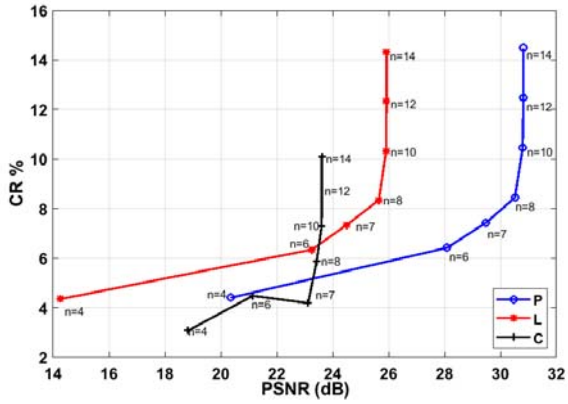



Figure 3. Performance for varying quantization levels on Pepper, P; Lena, L; and Cameraman, C.

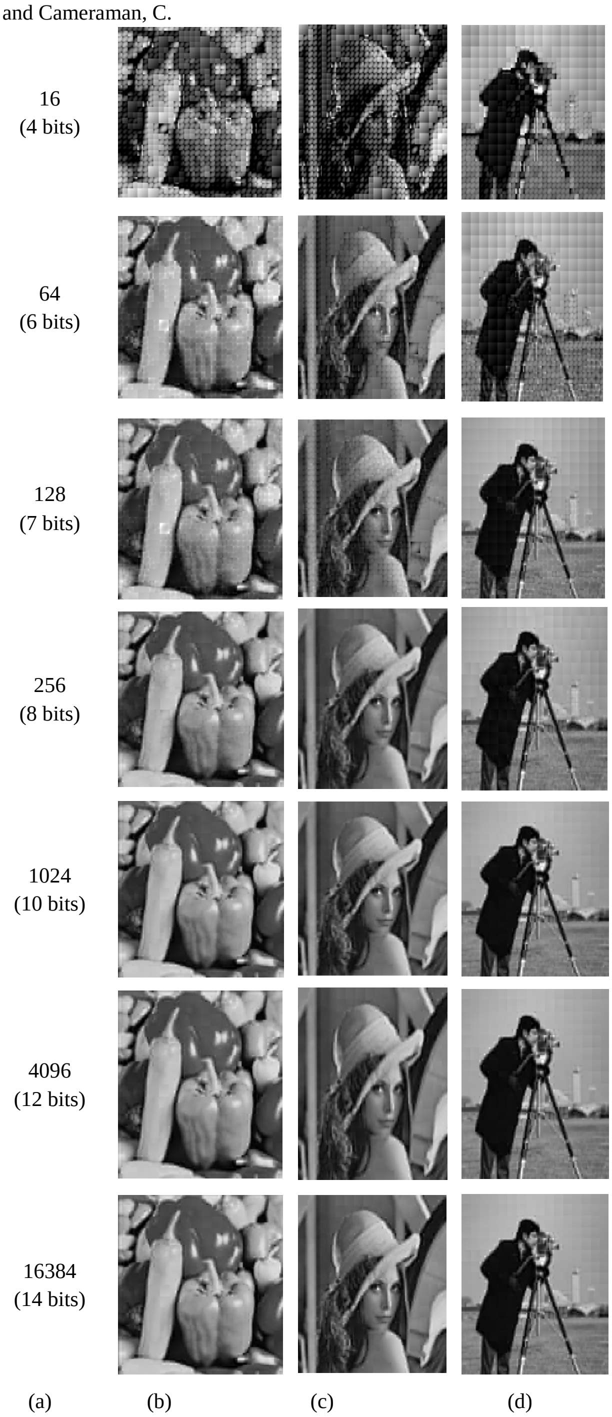



Figure 4. (a) Quantization levels (n bits); (b), (c), and (d) are the decompressed images for 4 to 14 bits for Pepper, P; Lena, L; and Cameraman, C, respectively.

### *C. Gray Image Experiment*

The used images in these experiments are Lena, Rice, and Cameraman. With this technique, we use 8x8 and 16x16 block sizes, a quantization level equal to 1024, and a threshold that varies from 1 to 40. Figure 5 shows the summary graph for the results of the proposed technique on gray test images at different threshold values, while Table II shows the best results of the proposed method for the three the best results of the proposed method for the three standard gray test images.

Figure 5 shows that the algorithm performance is better for Lena and Rice than for Cameraman. The reason for this difference in performance is the number of stable (minimal-variance) blocks and the sharp transitions in the blocks in Lena and Rice, which are greater than those in Cameraman. Additionally, from Table II, we note that our proposed method achieves a comparable result for the test images used.

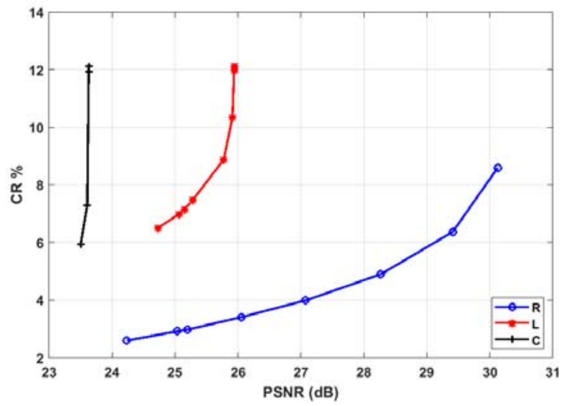



Figure 5. Summary graph for the results of the proposed technique on gray test images at different threshold values (Rice, R; Lena, L; and Cameraman, C)

### *D. Color Image Experiment*

The used images in these experiments are Pepper, Lena, Cameraman, and Kahramana. With our technique, we use 8x8 and 16x16 block sizes and a quantization level equal to 1024, and the threshold varies from 1 to 60. In our research, two-color spaces are used: RGB and YUV. RGB is the most common color space. YUV is used typically for color image processing. It encodes a color image and preserves the properties of the human eye, which allows a bandwidth reduction for chroma components without perceptual distortion. In our experiment, the original images are in RGB space. However, from our experimentations with different color spaces, we found that YUV gives better results which are consistent with other schemes such as JPEG.

Therefore, the original images are converted into YUV, then compressed, decompressed, and finally converted back into RGB to extract the performance metrics against the original images. The colored image is split into three channels. We apply the algorithm in each channel individually. Conversion equations between the two-color spaces are from Eq. (11) to Eq. (16), which are defined as [43].

From RGB to YUV,

$$Y=0.299R+0.587G+0.114B \tag{11}$$

$$U=-0.147R-0.289G+0.436B \tag{12}$$

$$V=0.615R-0.515G-0.100B \tag{13}$$

From YUV to RGB,

$$R=Y+1.140V \tag{14}$$

$$G=Y-0.395U-0.581V \tag{15}$$

$$B=Y+2.032U \tag{16}$$

Figure 6 shows the summary graph for the results of the proposed technique on the color test images at different threshold values. Table III shows the best results of the proposed approach for the four-color test images.

Figure 6 shows that good quality with the best CR is achieved for the Cameraman image, which includes fewer fine details and sharp edges.

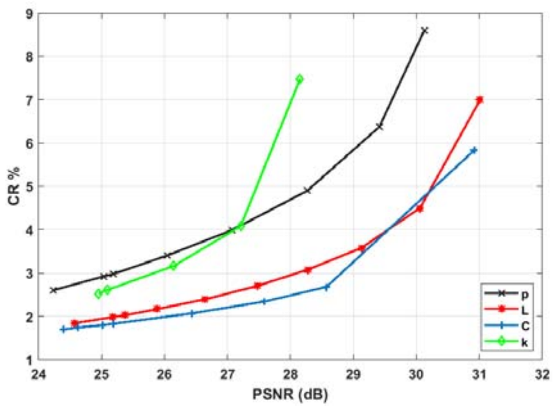



Figure 6. Summary graph for the results of the proposed technique on color test images at different threshold values (Pepper, P; Lena, L; Cameraman, C; and Kahramana, K)

## E. *Comparative Study*

Table IV shows a comparison between our results and the results obtained in [35] and [39]. We note that the proposed method outperforms the methods proposed in [35] and [39] for both gray and color images. Table V shows the differences between our approach and the approaches of [35] and [44]. We note that there are many differences between our approach and the other approaches in terms of the block sizes, adaptive polynomial order, encoding, and quantization methods.

## V. Conclusion and Future Work

In this paper, we propose a new lossy image compression encoding technique using polynomial curve fitting that represents many pixels with a minimal number of polynomial coefficients, which improves the overall performance. The proposed approach divides the images into blocks of two different sizes (16x16 and 8x8) and two polynomial degrees (first and second order). The choice of the block size and the polynomial degree depends on the threshold value. The fusion of polynomial curve fitting and the concept of adaptive blocks helps to represent a large number of pixels with few polynomial coefficients, which enhances the CR. To effectively store these coefficients, we use uniform quantization. The encoding process uses the blocks as they are, without transforming them to other domains. Moreover, it does not require stored encoding tables, and it does not dynamically build such tables. The decoding process is a simple single-path process, and it can be paralyzed. The proposed technique is applied to two different image types (gray and color). It can be concluded from the experimental results that the proposed approach improves the reconstruction quality (PSNR) and CR with a good structure similarity and minimum mean-squared error. The results are found to be comparable to those of previous methods. In future work, we plan to study the effect of dividing an image into four blocks and repeatedly dividing each block into four smaller blocks. Then, we will apply the wavelet transform to each block to show the effect of this method on both the CR and the quality of the reconstructed image. Additionally, we plan to use a biomedical image database for the performance test.

Table II. The Best Results of the Experiment on Gray Images

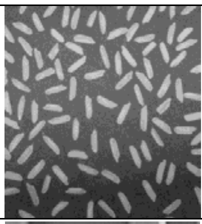
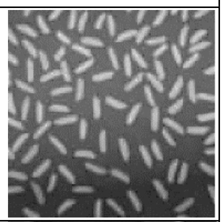
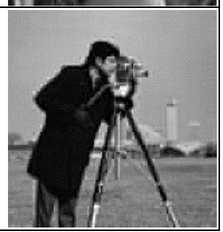
| **Image name** | **parameters** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Original image** | **PSNR (dB)** | **CR (%)** | **RMSE** | **SSIM** | **Encoding time (sec)** | **Decoding time (sec)** | **Decompressed image** |
| Rice |  | 32.8997 | 10.4786 | 5.775 | 0.79558 | 0.76201 | 0.091993 |  |
| Lena |  | 25.9113 | 10.3355 | 12.9114 | 0.77798 | 0.448 | 0.038 |  |
| Cameraman |  | 23.6291 | 11.9308 | 16.7913 | 0.76098 | 0.238 | 0.015993 |  |

TABLE III. THE BEST RESULTS OF THE EXPERIMENT ON COLOR IMAGES

| Image name | parameters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Original image** | **PSNR (dB)** | **CR (%)** | **RMSE** | **SSIM** | **Encoding time (sec)** | **Decoding time (sec)** | **Decompressed image** |
| Pepper |  | 30.1253 | 8.5991 | 7.9484 | 0.96927 | 0.528 | 0.057 |  |
| Lena |  | 31.0126 | 6.9977 | 7.1764 | 0.97378 | 0.40701 | 0.054003 |  |
| Cameraman |  | 30.9182 | 5.8334 | 7.2549 | 0.9759 | 0.363 | 0.052001 |  |
| Kahramana |  | 28.15 | 7.48 | 9.98 | 0.9353 | 0.269 | 0.031 |  |

TABLE IV. COMPARATIVE RESULTS OF OTHER METHODS AND THE PROPOSED METHOD FOR GRAY AND COLOR IMAGES

| Image type | Reference | Image name | PSNR (dB) | CR (%) | RMSE |
|---|---|---|---|---|---|
| **Gray** | [39] | Lena | 23.66 | 25 | 16.73 |
| | | Rice | 24.87 | 25 | 14.56 |
| | | Cameraman | 21.66 | 25 | 21.05 |
| | Proposed method | Lena | 25.91 | 10.34 | 12.91 |
| | | Rice | 32.90 | 10.48 | 5.775 |
| | | Cameraman | 23.63 | 11.93 | 16.79 |
| **Color** | [35] | Kahramana | 25.19 | 13 | 14.03 |
| | Proposed method | Kahramana | 28.15 | 7.48 | 9.98 |

TABLE V. A COMPARATIVE TABLE BETWEEN OTHER RESEARCHES AND OUR PROPOSED METHOD

| Parameter | [35] | [44] | Proposed method |
|---|---|---|---|
| **Block size** | Fixed block size | Fixed block size | Adaptive-block size. We use two block sizes: 16x16 and 8x8 |
| **Adaptive metric** | Block variance | Not applicable | RMSE |
| **Polynomial** | Adaptive polynomial | Fixed | Adaptive polynomial. We use two types of polynomials, of order 1 and order 2 |
| **Encoding** | Huffman encoding | No encoding | Adaptive encoding. Simple encoding method that does not require stored encoding tables |
| **Quantization** | Multiple levels of scalar quantization | Uniform quantization | Uniform quantization |

## REFERENCES


[1] H. Ackar, A. A. Almisreb, and M. A. Saleh, "A review on image enhancement techniques," Southeast Europe J. Soft Comput., vol. 8, no. 1, pp. 42–48, Apr. 2019. doi:10.21533/scjournal.v8i1.175

[2] M. Maru and M. C. Parikh, "Image restoration techniques: A survey," Int. J. Comput. Appl., vol. 160, no. 6, pp. 15–19, Feb. 2017. doi:10.5120/ijca2017913060

[3] N. Dhanachandra and Y. J. Chanu, "A survey on image segmentation methods using clustering techniques," Eur. J. Eng. Res. Sci., vol. 2, no. 1, pp. 15–20, Jan. 2017. doi:10.24018/ejers.2017.2.1.237

[4] P. V. Kumaraguru and V. J. Chakravarthy, "An image feature extraction and image representation strategies for the analysis of image processing," Indian J. Forensic Medicine Toxicology, vol. 11, no. 2, pp. 642–648, Nov. 2017. doi:10.5958/0973-9130.2017.00202.x

[5] K. Chopra and I. S. Virk, "Image steganography using edge detection technique," Int. J. Comput. Sci. Eng., vol. 6, no. 12, pp. 222–227, Dec. 2018. doi:10.26438/ijcse/v6i12.222227

[6] D. Maia and R. Trindade, "Face detection and recognition in color images under matlab," Int. J. Signal Process. Image Process. Pattern Recognit., vol. 9, no. 2, pp. 13–24, Feb. 2016. doi:10.14257/ijsip.2016.9.2.02

[7] M. A. Saleh, "Image steganography techniques - a review paper," IJARCCE, vol. 7, no. 9, pp. 52–58, Sep. 2018. doi:10.17148/ijarcce.2018,7910

[8] S. Khan and S. Khan, "An efficient content based image retrieval: CBIR," Int. J. Comput. Appl., vol. 152, no. 6, pp. 33–37, Oct. 2016. doi:10.5120/ijca2016911885

[9] M. Singh, S. Kumar, S. Singh, and M. Shrivastava, "Various image compression techniques: Lossy and lossless," Int. J. Comput. Appl., vol. 142, no. 6, pp. 23–26, May 2016. doi:10.5120/ijca2016909829

[10] N. S. Singh and H. J. Singh, "Data compression techniques in wireless sensor network a survey," Int. J. Comput. Sci. Eng., vol. 7, no. 1, pp. 697–706, Jan. 2019. doi:10.26438/ijcse/v7i1.697706

[11] A. J. Hussain, A. Al-Fayadh, and N. Radi, "Image compression techniques: A survey in lossless and lossy algorithms," Neurocomputing, vol. 300, pp. 44–69, Jul. 2018. doi:10.1016/j.neucom.2018.02.094

[12] S. Bhambay, S. Poojary, and P. Parag, "Fixed length differential encoding for real-time status updates," IEEE Trans. Commun., vol.

67, no. 3, pp. 2381–2392, Mar. 2019. doi:10.1109/tcomm.2018.2883423
[13] T. G. Swart and J. H. Weber, "Binary variable-to-fixed length balancing scheme with simple encoding/decoding," IEEE Commun. Lett., vol. 22, no. 10, pp. 1992–1995, Oct. 2018. doi:10.1109/lcomm.2018.2865350
[14] K. Negrat, R. Smko, and A. Almarimi, "Variable length encoding in multiple frequency domain steganography," in 2010 2nd Int. Conf. Softw. Technol. Eng., San Juan, PR, 2010, pp. V1-305–V301-309. doi:10.1109/icste.2010.5608853
[15] M. Jayedul and M. Nurul, "Study on data compression technique," Int. J. Comput. Appl., vol. 159, no. 5, pp. 6–13, Feb. 2017. doi:10.5120/ijca2017912416
[16] S. M. Hardi, B. Angga, M. S. Lydia, I. Jaya, and J. T. Tarigan, "Comparative analysis run-length encoding algorithm and fibonacci code algorithm on image compression," J. Phys., vol. 1235, p. 012107, Jun. 2019. doi:10.1088/1742-6596/1235/1/012107
[17] M. Sangeetha, P. Betty, and G. S. N. Kumar, "A biometrie iris image compression using LZW and hybrid LZW coding algorithm," in 2017 Int. Conf. Innov. Inf. Embedded Commun. Syst. (ICIIECS), Coimbatore, India, 2017, pp. 1–6. doi:10.1109/iciiecs.2017.8275906
[18] S. V. Viraktamath, M. V. Koti, and M. M. Bamagod, "Performance analysis of source coding techniques," in 2017 Int. Conf. Comput. Methodologies Commun. (ICCMC), Erode, India, 2017, pp. 689–692. doi:10.1109/iccmc.2017.8282554
[19] S. Yuan and J. Hu, "Research on image compression technology based on Huffman coding," J. Vis. Commun. Image Representation, vol. 59, pp. 33–38, Feb. 2019. doi:10.1016/j.jvcir.2018.12.043
[20] M. Sharma, "Arithmetic coding based string approximation," in 2017 6th Int. Conf. Rel. Infocom Technol. Optim. (Trends and Future Directions) (ICRITO), Noida, India, 2017, pp. 416–422. doi:10.1109/icrito.2017.8342462
[21] N. H. Salman, "New image compression/decompression technique using arithmetic coding algorithm," J. Zankoy Sulaimani Part A, vol. 19, no. 1, pp. 263–272, Oct. 2016. doi:10.17656/jzs.10604
[22] M. George, M. Thomas, and C. K. Jayadas, "A methodology for spatial domain image compression based on hops encoding," Procedia Technol., vol. 25, pp. 52–59, Dec. 2016. doi:10.1016/j.protcy.2016.08.080
[23] M. Alzain, "Image encryption using chaotic cat mapping in the discrete fourier transform," Int. J. Comput. Technol., vol. 18, pp. 7389–7397, Nov. 2018. doi:10.24297/ijct.v18i0.7907
[24] H. Ga and W. Zeng, "Image compression and encryption based on wavelet transform and chaos," Comput. Opt., vol. 43, no. 2, pp. 258–263, Apr. 2019. doi:10.18287/2412-6179-2019-43-2-258-263
[25] R. Shoitan, Z. Nossair, I. Isamil, and A. Tobal, "Hybrid wavelet measurement matrices for improving compressive imaging," Signal Image Video Process., vol. 11, no. 1, pp. 65–72, Apr. 2016. doi:10.1007/s11760-016-0894-5
[26] D. S. S. Satyanarayana, G. R. Reddy, R. S. P. Shanmukhanath, and K. Gaurav, "JPEG image compression using discrete cosine transform," Int. J. New Technol. Res., vol. 5, no. 10, pp. 52–55, Nov. 2019. doi:10.31871/ijntr.5.10.24
[27] S. Goswami and S. Mishra, "Lossless compression based image compression technique for medical imaging," Int. J. Recent Trends Eng. Res., vol. 4, no. 3, pp. 793–800, Sep. 2018. doi:10.23883/ijrter.2018.4377.lkmwe
[28] A. S. Yadav, H. Sadia, and S. Singh, "Lossy compression of color image using EZW and BTC," Int. J. Sci. Res., vol. 6, no. 12, pp. 1124–1127, Dec. 2017. doi:10.21275/art20176202
[29] T. Richter, "JPEG on STEROIDS: Common optimization techniques for JPEG image compression," in 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 61–65. doi:10.1109/icip.2016.7532319
[30] F. Artuger and F. Ozkaynak, "Fractal image compression method for lossy data compression," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1–6. doi:10.1109/idap.2018.8620735
[31] A. H. Abouali, "Object-based VQ for image compression," Ain Shams Engineering Journal, vol. 6, no. 1, pp. 211–216, Mar. 2015. doi:10.1016/j.asej.2014.10.007
[32] G. Y. Jiang, Y. P. Zhu, M. Yu, and Y. Zhang, "Perceptual video coding: A survey," J. Electron. Inf. Technol., vol. 35, no. 2, pp. 474–483, Feb. 2014. doi:10.3724/sp.j.1146.2012.00931
[33] Z. B. Wu and J. Q. Yu, "Vector quantization: A review," Frontiers of Information Technology and Electronic Engineering, vol. 20, no. 4, pp. 507–524, Apr. 2019. doi:10.1631/fitee.1700833
[34] A. Ghareeb and S. Z. Rida, "Image quality measures based on intuitionistic fuzzy similarity and inclusion measures," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 6, pp. 4057–4065, Jun. 2018. doi:10.3233/jifs-171480
[35] R. J. Al-Bahadili, "Adaptive polynomial fitting for image compression based on variance of block pixels," Engineering and Technology Journal, vol. 33, pp. 1830–1844, Mar. 2015.
[36] Y. Q. Cai, H. X. Zou, and F. Yuan, "Adaptive compression method for underwater images based on perceived quality estimation," Frontiers Inf. Technol. Electron. Eng., vol. 20, no. 5, pp. 716–730, May 2019. doi:10.1631/fitee.1700737
[37] S. Ameer and O. Basir, "Image compression using plane fitting with inter-block prediction," Image Vision Comput., vol. 27, no. 4, pp. 385–390, Mar. 2009. doi:10.1016/j.imavis.2008.06.005
[38] S. Sadanandan and V. K. Govindan, "Image compression with modified skipline encoding and curve fitting," International Journal of Computer Applications, vol. 74, no. 5, pp. 24–30, Jul. 2013. doi:10.5120/12882-9786
[39] S. Sajikumar and A. K. Anilkumar, "Image compression using chebyshev polynomial surface fit," International Journal Of Pure and Applied Mathematical Science, vol.10, pp. 15–27, Oct. 2017
[40] E. Erdal and A. Ergüzen, "An efficient encoding algorithm using local path on huffman encoding algorithm for compression," Applied Science, vol. 9, no. 4, p. 782, Feb. 2019. doi:10.3390/app9040782
[41] W. Khalaf, D. Zaghar, and N. Hashim, "Enhancement of curve-fitting image compression using hyperbolic function," Symmetry, vol. 11, no. 2, pp. 291, Feb. 2019. doi:10.3390/sym11020291
[42] Y. Zhang, H. Wang, and J. Zhao, "Depth map compression based on platelet coding and quadratic curve fitting," in 2014 IEEE 27th Canadian Conference of Electrical and Computer Engineering (CCECE), Toronto, ON, 2014, pp. 1–4. doi:10.1109/ccece.2014.6901002
[43] N. A. Ibraheem, M. M. Hasan, R. Z. Khan, P. K. Mishra," Understanding Color Models: A Review," ARPN Journal of Science and Technology, vol. 2, no.3,pp.265-275, Apr. 2012.
[44] S. M. Othman, A. E. Mohamed, Z. Nossair, and M. I. El-Adawy, "Image compression using polynomial fitting," in 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 344–349. doi:10.1109/iceca.2019.8822232