Adaptive Polynomial Fitting for Image Compression Based on Variance of Block Pixels

Dr. Riyadh Jabbar S. Al-Bahadili
Computer Engineering Department., University of Technology/Baghdad.
Email:dr_rjs2007@yahoo.com

Received on:25/12/2014 & Accepted on:30/8/2015

Abstract

This paper presents a proposed method to compress images using two polynomials with different models based on the value of block pixels variance. These two polynomials are chosen from different set of models, which give low number of coefficients and preserve the quality of image as much as possible. This procedure of adaptive fitting ensures that the number of coefficients for each block is as the minimum as possible depending on the value of block variance. After applying multi-level of scalar quantization and Huffman encoding to polynomials coefficients for each block of image and testing different variance thresholds; mean square error (MSE), peak signal to noise ratio (PSNR), processing time, and compression ratio (CR) are evaluated for two types of images (color and gray scales) and for different block sizes (4x4 and 8x8 pixels). Computer results showed that the proposed method gives an acceptable compression ratio and image quality compared with non-adaptive fitting. For 4x4-block size, there is an improvement in PSNR (25.19 dB) compared with nonlinear polynomial case (25.08 dB). In addition, CR (7.45) is better than both cases (7.11 for linear and 5.56 for nonlinear polynomial case). The results showed that the suggested method of adaptive polynomial fitting is more suitable for gray scale images (including handwriting images).

Keywords: Digital image processing, image compression, polynomial fitting, surface fitting

ملائمة متعدد الحدود المكيف لضغط الصور أعتمادا على تباين عناصر الصورة

الخلاصة

يقدم هذا البحث طريقة مقترحة لضغط الصور باستخدام دالتين من متعددات الحدود مع نماذج مختلفة بناء على قيم التباين لعناصر الصورة (pixels variance). تم اختيار متعددات الحدود هذه من مجموعة مختلفة من النماذج، والتي تعطي أقل عدد من المعاملات مع الحفاظ على جودة الصورة قدر الإمكان. هذا الإجراء من الملائمة المكيفة يضمن أن عدد المعاملات لكل كتلة من عناصر الصورة هو الحد الأدنى الممكن استخدامه اعتمادا على قيمة تباين الكتلة. بعد تطبيق مستويات متعددة من التكميم العددي (scalar quantization) وترميز هوفمان(Huffman encoding) لمعاملات متعددات الحدود لكل كتلة من الصورة واختبار عتبات مختلفة من تباين عناصر الصورة (variance thresholds) ، فأن معدل مربع

https://doi.org/10.30684/etj.2015.108826
2412-0758/University of Technology-Iraq, Baghdad, Iraq

الخطأ (MSE)، نسبة قيمة عنصر الصورة الى الضوضاء (PSNR)، وقت المعالجة، ونسبة ضغط الصورة قد تم حسابها لنوعين من الصور (ملون ورمادي المستويات) ولنوعين من أحجام كتل عناصر الصورة (4×4 و8×8 عنصر). أظهرت نتائج الحاسوب أن الطريقة المقترحة تعطي نسبة ضغط وجودة صوره مقبولة مقارنة مع طرق الملائمة الغير مكيفة لضغط الصور. في حالة حجم الكتلة 4×4 فقد تم الحصول على جودة صوره (21.19 دسيبل) مقارنة مع حالة متعدد الحدود الغير خطي (25.08 دسيبل). بالإضافة الى ذلك فأن نسبة الضغط في الطريقة المقترحة (7.45) هي أفضل من حالة متعدد الحدود الخطي (7.11) وحالة متعدد الحدود الغير خطي (5.56). أظهرت النتائج أيضا أن الأسلوب المقترح من ملائمة متعدد الحدود المكيف هو مناسب أكثر للصور ذات المستويات الرمادية (بما في ذلك الصور ذات الكتابة اليدوية).

INTRODUCTION

Due to growing of the visual information, it became very important to compress images or video, which leads to an efficient use of media storage and permits fast transmission. From the point of view of storage, gray scale images require 8-bit for each pixel, while color images requires 24-bit (three channels). For grayscale image (256x256), this means the uncompressed image requires 256x256x8 = 524288 bits, while 256x256x24 =1572864 bits for color image. Moreover, from the point of view of bandwidth, the size of uncompressed image requires high bandwidth with low transfer rate. That is why the compression process plays an important role in dealing with multi media. Recently, many algorithms for image compression have been developed which based on block, pixel, band, or region of image, which depended on the fact that there is high correlation among neighboring image pixels values. This property of high-correlation means any image that contains high redundant information [1]. These techniques of image compression are either reversible (lossless) or irreversible (some of image information are lost) process.

- In lossless compression algorithms, the original image is recovered from the compressed image without losses any information. They use statistical methods to minimize (or eliminate) the redundancy [2]. Most of applications that need accurate requirements such as medical imaging use lossless compression techniques. These techniques include run length encoding [3], Huffman encoding [4], Lempel–Ziv–Welch (LZW) coding [5], and area coding [6].
- Lossy compression algorithms give higher compression ratios than lossless algorithms. They are widely used in most applications when the quality of images is not the mean issue. These techniques include: transformation coding (such as discrete Fourier transform (DFT), discrete cosine transform (DCT), and discrete wavelet transform (DWT)) [7-9], vector quantization, fractal coding, block truncation coding (BTC), and subband coding [10].

Neural networks have also been used for compressing process, but they have low compression ratio [11]. A compression technique has also been proposed which combines fuzzy logic with Huffman coding [12] but required high processing time. Other compression techniques are Interpolation and surface fitting. All image interpolation methods involve fitting missing pixels to some sample structure learnt from the low-resolution image [13]. Surface fitting method uses single polynomial (first-order or second order) that fits the values of blocks pixels and transmits only the coefficients of the polynomial for each block [14, 15]. Regardless of compression algorithm used, the compression process in general tries to reduce the size of uncompressed image as minimum as possible while maintaining the quality of reconstructed image as closed as original image. Various parameters are used to test the performance of image compression process such as mean squared error (MSE), peak signal to noise ratio (PSNR), and compression ratio (CR). For an *mxn* image, the MSE is the cumulative squared error between uncompressed (original) image $f(i,j)$ and the reconstructed (approximated version) image $g(i,j)$ where $i = 1,2 \ldots m$ and $j = 1,2 \ldots n$ and defined by [16]:

$$\mathrm{MSE} = \frac{1}{\mathrm{mxn}} \sum_{i=1}^{m} \sum_{j=1}^{n} |f(i,j) - g(i,j)|^2 \qquad \ldots (1)$$

PSNR is also used as a measure of quality of reconstructed image and defined as:

$$PSNR = 20\, log_{10} \left[\frac{2^b - 1}{\sqrt{MSE}}\right] \qquad \ldots (2)$$

Where
b is pixel depth in bits (for grayscale, $b = 8$ bits). Low MSE and high PSNR means better compression scheme. Another parameter is compression ratio, which is defined as:

$$CR = \frac{uncompressed\ image}{compressed\ image} \qquad \ldots (3)$$

In this paper, a proposed method uses surface fitting adaptively to compress the image. Instead of single polynomial, which is used in the traditional surface fitting, two polynomials will be used for whole image while just one of them is used for each block depending upon the variance of pixels values of that block.

Non-adaptive surface fitting

In surface fitting (or polynomial fitting), it is considered that image information in 2D matrix form, and be constructed of *m* rows and *n* columns. Each element of this matrix represents the pixel value (from 0 to 255 in grayscale). This image is divided to non-overlapping blocks (such as 4x4 or 8x8 pixels), and the following polynomial applies to each block [17].

$$P(x,y) = p_0 + p_1 x + p_2 y + p_3 x^2 + p_4 y^2 + \cdots + p_{s-1} x^k + p_s y^k \quad \ldots (4)$$

Where
$p_0, p_1, \ldots, p_s$ are the polynomial coefficients and k is the order of polynomial. The coefficients are calculated such that the MSE is minimized for each block. A simplified application of first order polynomial fitting was used in [14]. After extracting the coefficients, the quantization and coding process (such as Huffman coding) are applied to minimize the bits-representation of coefficients. High-order polynomial means low MSE and gives good image quality. However, in the same time gives low compression ratio due to increasing in the number of coefficients. In the case of low-order polynomial (such as first-order) gives high compression ratio but low image quality. In general, fitting by

high-order polynomials increases the time processing to compress the image, compared to the first-order polynomial. This paper will test the following two polynomials (first-order and second-order) for comparison with the proposed method.

$$P(x,y) = p_0 + p_1 x + p_2 y \qquad \dots (5)$$

$$P(x,y) = p_0 + p_1 x + p_2 y + p_3 x^2 + p_4 y^2 \qquad \dots (6)$$

Both polynomials will apply separately to the same color image (Kahramana 128x256) for different block sizes (4x4 and 8x8) as in the following steps:
Step.1: the splitting the color image into three channels (red, green, and blue) as shown in Figure.1.
Step.2: For each channel, the matrix is divided into 4x4 blocks.
Step.3: For each block, first-order polynomial (eq.5) is applied and the three coefficients are calculated.
Step.4: These coefficients (except p_0) are quantized using uniform quantizer which has 2^5 levels (which means each coefficient will take 5 bits).
Step.5: For additional minimization of the bit-representation of coefficients, Huffman coding is applied to encode the coefficients values according to the probability of occurrence of the coefficient.
The same steps are applied for block size 8x8 and for second-order polynomial (eq.6). For decompression process, each block is constructed (16 points for case 4x4-block) from its own coefficients and the final image is reconstructed from all blocks. The compression time is evaluated for each case and the parameters (eqs.1-3) are calculated using MATLAB (R2014a) program.

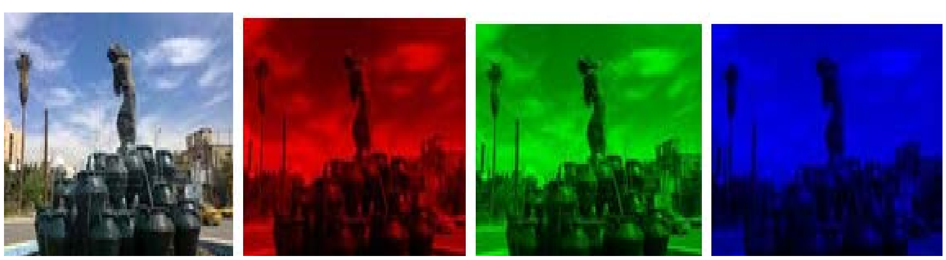
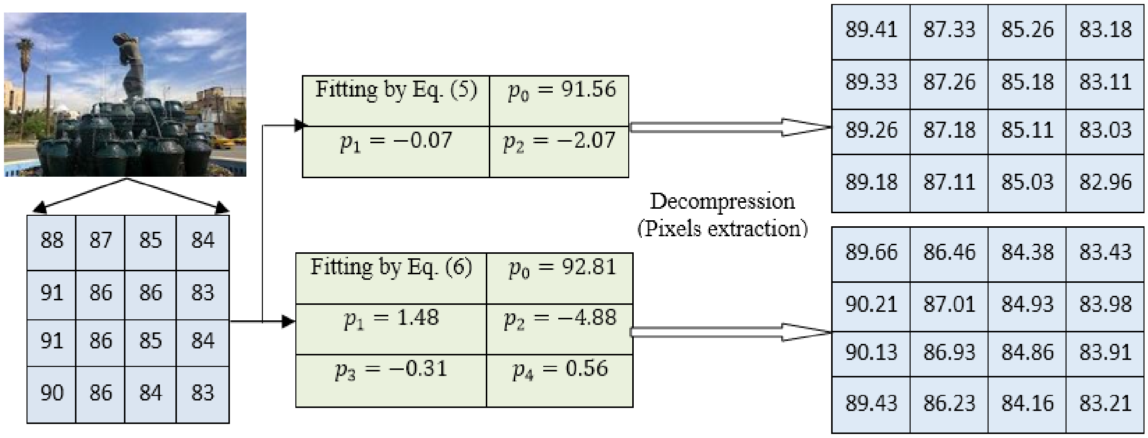
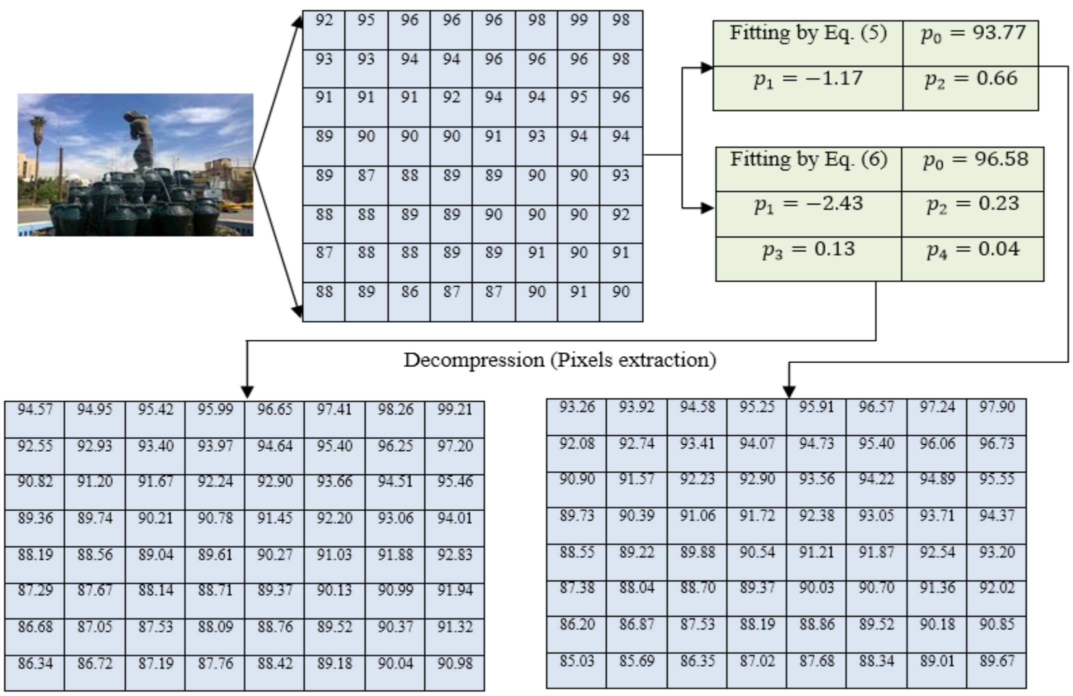


Figure.(1) original image (Kahramana 128x256) splitting into three channels

Sample calculations for coefficients of first-order and second-order polynomials (eq.5 and eq.6) and for both cases 4x4 and 8x8 block size, can be shown in Figure.2 and Figure.3 for randomly selected blocks.



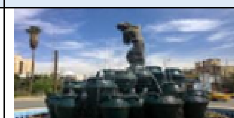
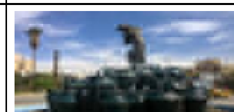
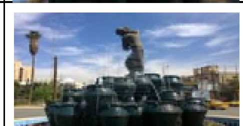
Figure(2) Polynomial fitting and pixels extraction for 4x4 block size



Figure(3) Polynomial fitting and pixels extraction for 8x8 block size

The results can be shown in Table.1.

Tabele(1) First and Secondorder Surfaces Fitting

Polynomial	Block Size	MSE	PSNR (dB)	CR	Compression Time per Blk (sec)	Reconstructed Image
First-order (eq.5)	4x4	225.81	24.59	7.11	0.018	
	8x8	474.58	21.36	28.44	0.036	
Second-order (eq.6)	4x4	134.10	26.85	4.26	0.042	
	8x8	371.05	22.43	17.06	0.045	

The results in Table.1 show clearly that there is a big gap between compression ratio and the quality of image. In the case of first-order polynomial and 8x8-block size, there is high compression ratio (about 28.44) but at the expanse of image quality (PSNR=21.36 dB). While in the case of second-order polynomial and 4x4-block size used, the image quality is more acceptable (PSNR=26.85 dB) but has less compression ratio (about 4.26).

Adaptive polynomial fitting (proposed method)

It is clear that there is a big gap between the compression ratio and the quality of image in the traditional surface fitting. The reason of this gap is that the non-adaptive surface fitting uses only a single polynomial for the whole blocks of the image regardless what the type of each block (high correlated or low correlated). In the proposed method, compression ratio and the quality of image will be compromised by using two different polynomials depending on the value of block variance. The flow chart for compression process of the proposed method can be shown in figure.4

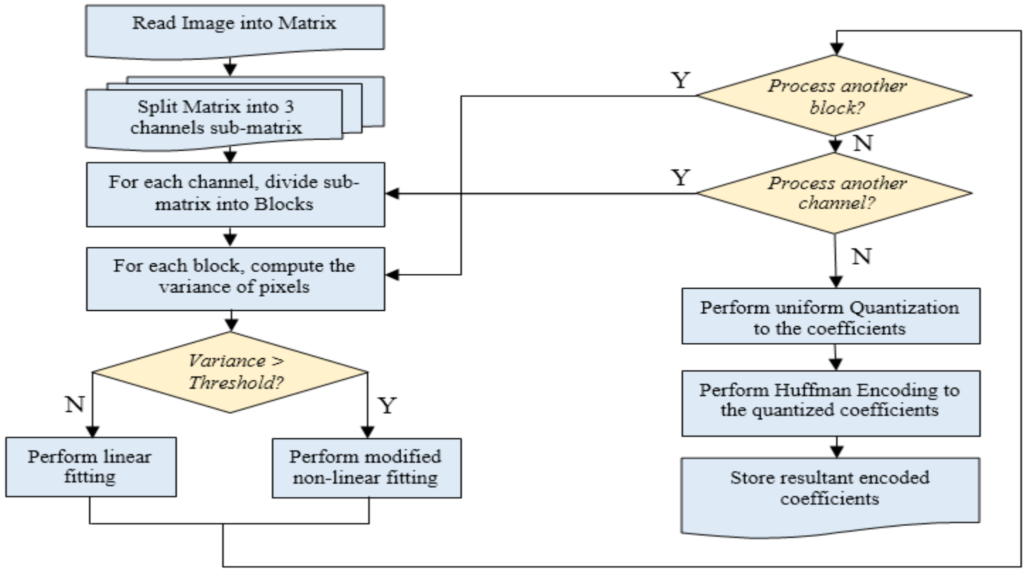


Figure (4) Compression process of the proposed method

Selection of polynomials

The proposed method in this paper will treat the blocks of the image separately depending on the variance of the block. Any block that has low variance, will be fitted using the standard first-order polynomial (eq.5). While any block that has high variance, will be fitted by the following modified polynomial:

$$P(x, y) = p_0 + p_1x + p_2y + p_3xy \quad \dots (7)$$

This type of modified polynomial avoids using high-order (and hence avoids more coefficients used) just to keep the compression ratio at acceptable level. The term (p_3xy) was added to the polynomial (eq.5) just to give the fitting process some of non-linearity behavior. This non-linearity leads to an improvement in reconstructed image quality for the blocks that have high variance values (at the expanse of increasing the number of coefficients by one compared with the first polynomial).

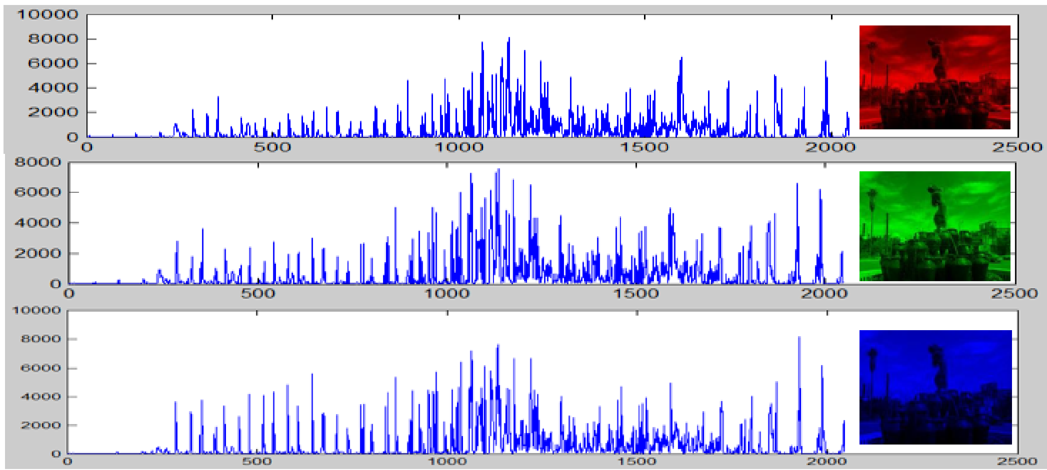
Selection of variance threshold

The variance (σ^2) of pixels values in each block is defined as the average of the squared differences from the mean and calculated as following [18]

$$\sigma^2 = \frac{\sum_{i=1}^{N} |f_i - \mu|^2}{N} \quad \dots (8)$$

Where

f_i is *i-th* pixel value (0 ~ 255), μ is the mean value of all pixels values in the block, and *N* is the block size (16 for the 4x4-block, and 64 for the 8x8-block). Figure.5 shows the variance (y-axis) of all 4x4 blocks (x-axis is block number) in all channels (R, G, and B) in the original image (Kahramana 128x256)



Figure(5) Variance of all channels in original image

In Figure.5, the number of 4x4 blocks in each channel is 2048. The maximum value of the variance is 8159 and the minimum value is 0.1625. It is clear that the variance is low when the pixels in a block are high-correlated and the variance is high when the pixels are low-correlated. As mentioned in previous section the selection of type of polynomial (linear or non-linear) will be based on the value of variance of the processed block. This block variance must be compared with a reference value (threshold). Selecting low threshold means the majority of blocks will be fitted by non-linear polynomial and hence results more coefficients and lower CR, while selecting high threshold means the majority of blocks will be fitted by linear polynomial and hence results lower image quality. Therefore, the optimum threshold in the proposed method will be selected to get a trade-off between CR and image quality. Table.2 shows the effect of choosing the threshold on the values of MSE, PSNR, and CR.

Table (2) Selection of Variance Threshold for 4X4 blocks

Threshold	Av. No. of linear blocks/Ch.	Av. No. of non-linear blocks/Ch.	No. of total coefficients/Ch.	MSE	PSNR(dB)	CR
0	0	2048	8192	201.492	25.088	4.010
1	64	1984	8128	201.492	25.088	4.029
2	130	1918	8062	201.494	25.088	4.068
3	190	1858	8002	201.498	25.088	4.097
10	382	1666	7810	201.540	25.087	4.205
20	515	1533	7677	201.622	25.085	4.292
30	621	1427	7571	201.716	25.083	4.358
40	649	1399	7543	201.802	25.081	4.403
50	762	1286	7430	201.888	25.079	4.443
60	831	1217	7361	201.967	25.078	4.480
70	886	1162	7306	202.055	25.076	4.511
80	929	1119	7263	202.138	25.074	4.540

Selection of quantization levels

The effect of selection of quantization levels (like the selection of variance threshold) will control the image quality and CR. More quantization levels means better quality but low CR, while less quantization levels means high CR but at the expanse of image quality. Table.3 shows the behavior of the coefficients for randomly selected blocks from the previous test (Table.2).

Table (3) Randomly Selected Coefficients

Coefficients of linear polynomial $P(x,y)=p_0+p_1x+p_2y$			Coefficients of non-linear polynomial $P(x,y)=p_0+p_1x+p_2y+p_3xy$			
p_0	p_1	p_2	p_0	p_1	p_2	p_3
219.0000	-0.5500	1.7000	226.7500	-3.6500	-1.4000	1.2400
225.3125	-2.1250	2.0250	233.0000	-5.2000	-1.0500	1.2300
233.3750	-3.0000	0.2000	240.7500	-5.9500	-2.7500	1.1800

In Table.3, it is clear that the first coefficient (p_0) in both polynomials has high priority than the rest of coefficients. So the coefficient p_0 will be truncated by only removing the four LSB during compression process then padding four zeros in decompression process, while others coefficients (which have smaller values) will be rounded and quantized by using 5-bits. The step size (Δ) between any two-quantization levels is evaluated according to the following equation [18]:

$$\Delta = \frac{|Max\ Coefficient - Min\ Coefficient|}{2^5} \qquad \dots (9)$$

Applying Huffman encoding

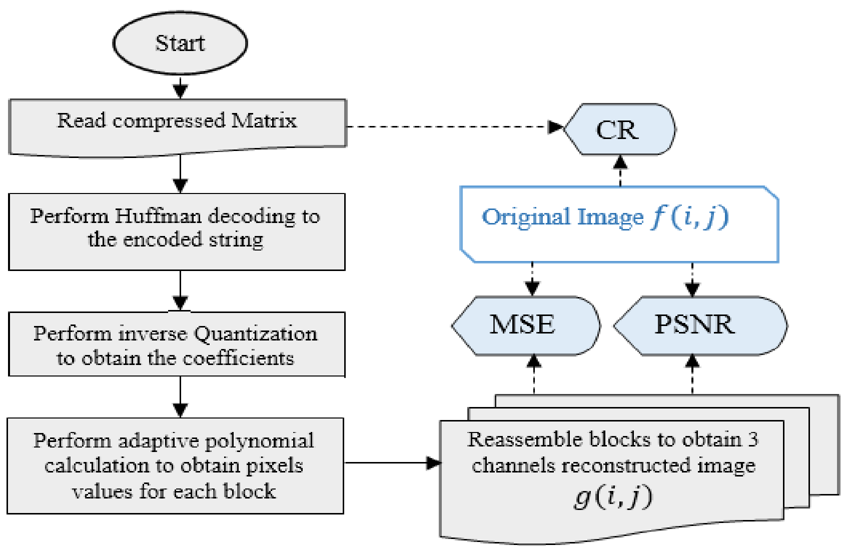
To increase CR further, Huffman encoding algorithm will be applied to the quantized coefficients. This process will not affect the image quality; it just minimizes the bit representation of the quantized coefficients. The Huffmann encoding is an optimum coding in the sense that no other uniquely decodable set of code words has a smaller average code-word length for a given source. The Huffmann encoding algorithm works as follows [18]:

1. The quantized coefficients are arranged in the descending order
2. The quantized coefficients of least probabilities are regarded as being combined into a new symbol with probability equal to the sum of the two original probabilities. The probability of the new symbol is placed in the list in accordance with its value.
3. The procedure is repeated until the final list of symbols. Symbol of only two for which a ‘0’ and a ‘1 ‘are assigned. The code for each symbol is found by working backward and tracing the sequence of 0s and 1s assigned to that symbol as well as its successor. Figure.6 shows a simple example to encode 5 quantized coefficients.

quantized coefficient	Step 1 Prob.(Ci)	Final Codes	Step 2		Step 3		Step 4	
C1	.4	00	.4	00	.4	1	.6	0
C2	.2	10	.2	01	.4	00	.4	1
C3	.2	11	.2	10	.2	01		
C4	.1	010	.2	11				
C5	.1	011						

Figure (6) Huffman encoding Process

In decompression process, every symbol in coded string can be decoded by examining this string from right to left and obtaining the original quantized coefficients. Figure.7 shows the block diagram of the decompression process for proposed method.



Figure(7) Block diagram of Decompression process for the proposed method

Results

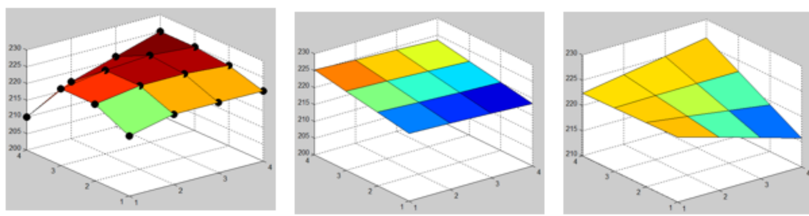
MATLAB (R2014a) m-files implementations have been carried out for both types of images color and grayscale (including handwriting image) and for two different block size (4x4 and 8x8). These results are compared with non-adaptive surface fitting method based on MSE, PSNR, and CR. The model of linear polynomial (eq.5) used MATLAB function library (poly11), while the model of modified non-linear polynomial (eq.7) used a custom model as shown in Table.4

Table (4) Polynomials models and related Matlab functions

Polynomial Type	MATLAB Function
$P(x,y) = p_0 + p_1x + p_2y$	$surffit = fit([x,y], z, 'poly11')$
$P(x,y) = p_0 + p_1x + p_2y + p_3xy$	*P=fittype (@ (a, b, c, d, x, y) a+b*x+c*y+d*x.*y, 'independent', {'x', 'y'} ,'dependent', 'z')* $surffit = fit([x,y], z, P, 'StartPoint', [1,1,1,1])$

Figure.8 shows the behavior of both polynomials (eq. 5 and 7) for randomly selected 4x4-block of the original image.

218	221	221	221
223	225	225	224
223	225	226	225
210	217	221	225



(a) (b) (c) (d)

Figure (8) shows the behaviore of linear and non-lineare polynomials (a) randomly selected 4X4 –block,(b) 3D Surface of original pixels, (c) fitting of Linear polynomial, (d) fitting of non-linear polynomial

As mentioned in section 3.2, the optimum variance threshold will be selected to get a trade-off between CR and image quality. Figure.9 shows the working range of variance threshold for the data in table.2.

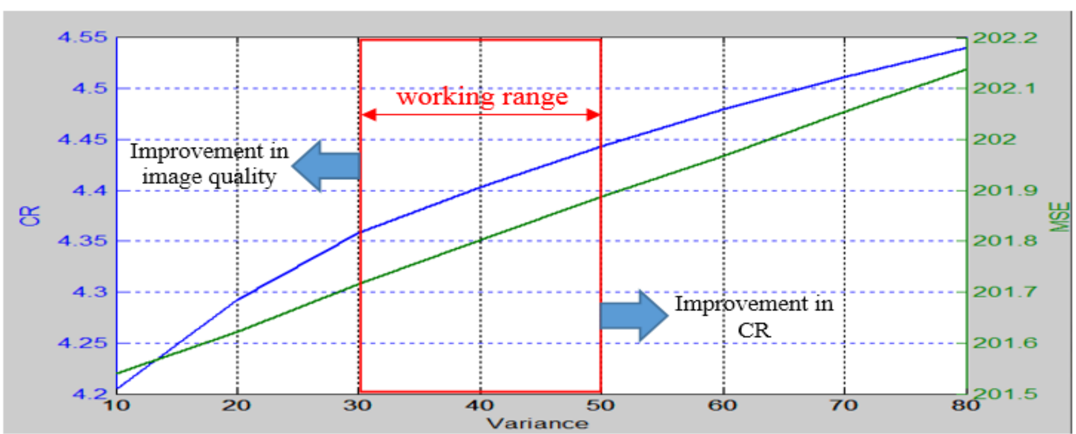


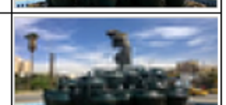
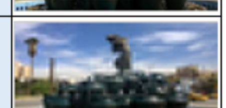
Figure (9) the working range of variance threshold

It is clear that there is a compromise between CR and MSE. Low threshold can be used for applications that seek for high image quality, while high threshold can be used for applications that seek for high compression ratio. Table.5 shows the results of non-adaptive fitting and the proposed method against JPG compression method for the following parameters:

- Color Image 256x128 (uncompressed size=256*128*3*8=786432 bits).
- Non-overlapped 4x4 and 8x8 block sizes.
- The value 40 has been taken as a variance threshold.
- 5-bit uniform quantization for coefficients p_1 , p_2 and p_3 has been performed using MATLAB function *imquantize ( )*, while coefficient p_0 in proposed method has been truncated by removing the four LSB during compression process then padding four zeros in decompression process.
- The Huffman encoding and decoding have been performed using the functions mat2huff ( ) and huff2mat ( ) which have been described in [16].
- The calculation of processing time for a block compression has been performed by using the MATLAB commands (tic-toc).

It is clear that from Table.5 the proposed method has a trade-off between PSNR and CR. For 4x4-block size, an improvement in PSNR (25.19 dB) is compared with nonlinear polynomial case (25.08 dB).

Table.(5) Results of non-adaptive fitting and proposed method against JPG compression

Method	Compression Type	Blk Size	MSE	PSNR (dB)	Compression size (bit)	CR	Compression Time per Blk (sec)	Reconstructed Image
Non adaptive fitting	Linear polynomial (eq.5)	4x4	225.81	24.59	110592	7.11	0.015	
		8x8	474.58	21.36	27648	28.44	0.031	
	Nonlinear polynomial (eq.7)	4x4	201.49	25.08	141312	5.56	0.034	
		8x8	437.29	21.72	35328	22.26	0.048	
Proposed method	Adaptive polynomial	4x4	196.80	25.19	105481	7.45	0.029	
		8x8	437.46	21.71	27559	28.53	0.045	
JPG	DCT	8x8	63.53	30.13	130419	6.03	0.00046	

Even the CR (7.45) is better than both cases (7.11 for linear and 5.56 for nonlinear polynomial). For 8x8-block size, high CR (28.53) has been achieved in proposed method compared with non-adaptive and JPG methods but there is no improvement in image quality (JPG method stills has high image quality).

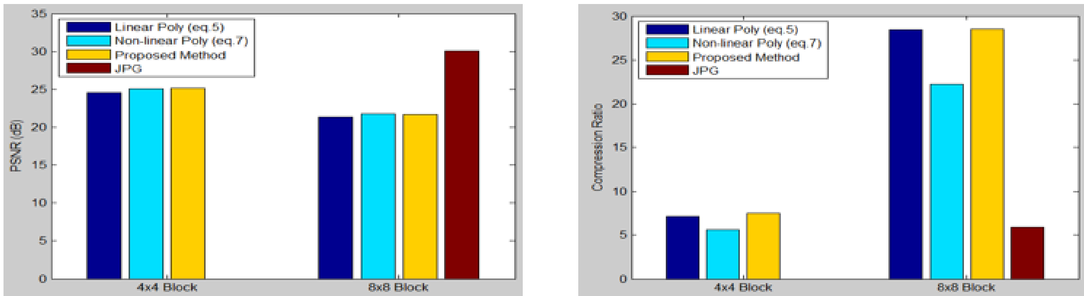


Figure.(10) PSNR and CR comparison for proposed method with non-adaptive polynomial fitting and JPG methods

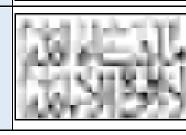
JPG method gave high image quality in spite of its CR (6.03) is far less than proposed method CR (28.53). From these results, it appears that the proposed method is more suitable for 4x4-block than 8x8-block for acceptable image quality, unless high CR is required. The comparison of results can be shown clearly in Figure.10 for PSNR and CR. The final test will be performed with grayscale and handwriting images to show the performance of proposed method when compresses such these types of images. Table.6 shows the results for this test for grayscale image with 4x4-block size and for different Thresholds values.

Table.(6) Results of Proposed method for gray scale image

Image Type	Image Preview	Variance Threshold	Compressed size (bit)	MSE	PSNR (dB)	CR	Reconstructed Image
(128x256) Gray Scale 262144 bits		40	37332	216.22	24.78	7.02	
		500	32107	224.07	24.62	8.16	
		1000	30447	232.50	24.46	8.61	

It seems from results in Table.6 that for seeking high CR, threshold value should be increased without degrading the image quality (details of the gray scale image can still be seen clearly). Finally, Table.7 shows the results for handwriting image with variance threshold 40 and for different block sizes.

Table.(7) Results of prposed method for handwriting image

Image Type	Image Preview	Block Size	Compressed size (bit)	MSE	PSNR (dB)	CR	Reconstructed Image
(128x256) Handwriting 262144 bits		4x4	34517	394.66	22.16	7.59	
		8x8	9178	2.04e+3	15.01	28.56	
		16x16	2417	4.49e+3	11.60	108.45	

Clearly, the 8x8-block is a better choice for high compression ratio (28.56) with an acceptable image quality for reading the text inside it. For 16x16-block, CR is very high (108.45) but has very poor quality (despite the fact the text inside it could be a little recognized).

Conclusion

In this paper, an adaptive polynomial fitting is proposed for image compression that based on the value of block-pixels variance. When combining with uniform quantizer and Huffman encoding, the proposed method provided a compromise between compression ratio and image quality due to the adaptation in suitable polynomial selection. Both polynomials that have been used, avoided using high-order (and hence avoided more coefficients used) just to keep the compression ratio at an acceptable level without degrading the image quality. Matlab results showed that the adaptive polynomial fitting is better than non-adaptive one in all types of images. Comparing with JPG method, the proposed method achieved high compression ratio (7.45) for 4x4-block size comparing with 8x8-block size JPG compression ratio (6.03) but with a little degradation of quality of image. Moreover, testing handwriting image showed high compression ratio (about 28.56) with an acceptable image quality for reading the text inside it. Even with 16x16-block size, the compression ratio reached to (108.45) but the text still hardly readable and needs some denoising filters for image enhancement.

References

[1] Subramanya A, “Image Compression Technique,” Potentials IEEE, Vol. 20, Issue 1, pp 19-23, 2001.

[2] Ming Yang & Nikolaos Bourbakis ,"An Overview of Lossless Digital Image Compression Techniques" Circuits & Systems, 2005 48th Midwest Symposium ,vol. 2 IEEE ,pp 1099-1102,7 – 10 Aug, 2005.

[3] Debashis Chakraborty et al., "Efficient Lossless Colour Image Compression Using Run Length Encoding and Special Character Replacement", International Journal on Computer Science and Engineering, Vol. 3 No. 7 July 2011.

[4] Mamta Sharma, "Compression Using Huffman Coding", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.

[5] Simrandeep kaur, V.Sulochana, "Design and Implementation of LZW Data Compression Algorithm", International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.4, July 2012.

[6] Deepa Raj, "Modeling of Image Compression and Decompression using Huffman Code Technique", International Journal of Computer Science and Mobile Computing, Vol.2, Issue 12, 2013.

[7] H. B. Kekre, Tanuja Sarode, Prachi Natu, "Image Compression Using Real Fourier Transform, Its Wavelet Transform And Hybrid Wavelet With DCT", International Journal of Advanced Computer Sciences and Applications, Volume: 4 Issue: 5 Page: 41-47, Science and Information Society (SAI), Iraqi Virtual Science Library, 2013.

[8] Salih H. A., Aymen D. S.," Image Compression Based on 2D Dual Tree Complex Wavelet Transform", Engineering & Technology Journal / University of Technology, Vol 28, No 7, 2010.

[9] Yasir A. A., Hadeel N. A., " Image Compression Using Lifting Scheme" , Engineering & Technology Journal / University of Technology, Vol 28, No 17, P. 5467-5455, 2010.

[10] Dr. Ajit Singh, Meenakshi Gahlawat, "Image Compression and its Various Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.

[11] I. Vilovic, "An experience in image compression using neural networks," in 48th International Symposium ELMAR-2006, Zadar, Crotia, 07-09, pp.95–98. , June 2006.

[12] R.K. Gangwar et al, "Performance analysis of image compression using fuzzy logic algorithm", Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.2, 2014

[13] Xiangjun Zhang and Xiaolin Wu, "Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation", IEEE transactions on image processing, vol. 17, No. 6, JUNE 2008.

[14] Amar Majeed Butt, Rana Asif Sattar, "On Image Compression Using Curve Fitting", Master Thesis, Computer Science, Thesis No. MCS-2010-35, 2010.

[15] Y.S. Chen, H.T. Yen, W.H. Hsu, "Compression of Color Image via the Technique of Surface Fitting", Graphical Models and Image Processing, Volume 56, Issue 3, Page 272–279, 1994.

[16] R.C. Gonzalez, "Digital image processing using Matlab", Pearson Prentice Hill, 2004.

[17] J.H.Mathews, K.D.Fink, "Numerical methods using Matlab", fourth edition, Pearson prentice hall, 2004.

[18] J.S.Chitode, "Information Coding Techniques", 1st edition, 2008.