

Article

Pre and Postprocessing for JPEG to Handle Large Monochrome Images

Walaa Khalaf ^{*,†} , Abeer Al Gburi [†]  and Dhafer Zaghar [†]

Computer Engineering Department, College of Engineering-Mustansiriyah University, Baghdad 10047, Iraq; akengineer2015@uomustansiriyah.edu.iq (A.A.G.); drz.raw@uomustansiriyah.edu.iq (D.Z.)

* Correspondence: w.khalaf@uomustansiriyah.edu.iq; Tel.: +964-7711141640

† These authors contributed equally to this work.

Received: 12 October 2019; Accepted: 26 November 2019; Published: 1 December 2019



Abstract: Image compression is one of the most important fields of image processing. Because of the rapid development of image acquisition which will increase the image size, and in turn requires bigger storage space. JPEG has been considered as the most famous and applicable algorithm for image compression; however, it has shortfalls for some image types. Hence, new techniques are required to improve the quality of reconstructed images as well as to increase the compression ratio. The work in this paper introduces a scheme to enhance the JPEG algorithm. The proposed scheme is a new method which shrinks and stretches images using a smooth filter. In order to remove the blurring artifact which would be developed from shrinking and stretching the image, a hyperbolic function (tanh) is used to enhance the quality of the reconstructed image. Furthermore, the new approach achieves higher compression ratio for the same image quality, and/or better image quality for the same compression ratio than ordinary JPEG with respect to large size and more complex content images. However, it is an application for optimization to enhance the quality (PSNR and SSIM), of the reconstructed image and to reduce the size of the compressed image, especially for large size images.

Keywords: image compression; JPEG; hyperbolic function; PSNR; SSIM; compression ratio; optimization

1. Introduction

In recent years, image compression has been considered as an attractive research field. Frequently, data are represented using large size images such as wallpaper and high quality media, which in turn need to be stored and transmitted without requiring large storage space or increased transmission rate of the communication channel [1]. In general, image compression with better quality reconstructed images is the main goal of any compression technique. This involves removing the redundancy and minimizing the loss in the image [2].

Image compression algorithms can be categorized into either lossless or lossy [1,3]. While lossless compression methods conserve the original image to be recovered completely after the decompression process [4], lossy compression uses the inherent redundancies found in an image, such as inter-pixel redundancy, psycho-visual redundancy, or coding redundancy, to decrease the data amount needed to represent the image [5–7]. Accordingly, lossless methods produce low compression ratio and error-free images; meanwhile, lossy methods produce high compression ratio with additional error (PSNR) [8].

Image compression is implemented into spatial domain and frequency domain. In spatial domain, image compression techniques aim to reduce the number of pixels representing the image without influencing the quality of the resulted image [9–11]. In frequency domain, Discrete Cosine Transform (DCT) [12,13], Discrete Fourier transform, or Discrete Wavelet Transform [5,14,15] are used to represent the energy of the image into a few number of coefficients.

JPEG is the most widely used method for lossy compression of digital photographs. Other sophisticated popular standards are JPEG2000, WebP, and Better Portable Graphics (BPG) [16]. In the JPEG process, an image is divided into several 8×8 blocks. Then, two-dimensional Discrete Cosine Transform (2D DCT) is applied for encoding each block. After performing DCT, most energy is concentrated in the low frequency region, which is very beneficial for compression as the human eye is sensitive to it. Subsequently, quantization is carried out for each block, where all 64 coefficients are quantized according to the desired image quality. Certain lossless compression operations are performed on the quantized data, consisting of a zig-zag scan of coefficients and entropy coding. The results of this process are rounded to integers. As a result of this step, some of the image information is lost. Finally, the Huffman method is used to encode the reduced coefficients using Huffman codes [17].

Although JPEG is a very widely used standard for image compression, it is still not applicable for many types of images like hyperspectral, radar, and medical images. Most objects are irregularly shaped and are not well approximated by the combination of rectangular blocks. In the block encoding process, a number of undesirable artifacts are introduced in the image, such as blocking artifacts (caused by discontinuities at the block boundaries) and ringing artifacts (caused by oscillations due to the Gibbs phenomenon). These problems become more apparent with increasing compression ratio [17,18].

Various studies have been carried out to improve and outperform well-known lossy compression methods, especially JPEG [19,20]. The aim is to make degraded images perceived better visually. This is a fundamental problem in the image processing field and is a subjective issue, as the quality of an image is decided by Human Visual System (HVS). In terms of the decompressed image, these studies aim to achieve better brightness and contrast, good color consistency, reduced noise, and better resolution than well-known lossy image compression methods. One way to improve the compression quality is to denoise the image as a pre-processing step using smoothing filters and median filters [19]. Another method is to reapply JPEG by associating the image database [17,18,20]. In [17], shape adaptive image compression algorithms were addressed. Here, the Shape-Adaptive Discrete Cosine Transform (SA-DCT) is used for transforming and encoding each block. The paper generalizes the JPEG algorithm and divides an image into trapezoid and triangular blocks according to the shapes of objects to achieve higher compression ratio. As it had replaced the use of 8×8 blocks adaptively with triangular, trapezoid, and polygonal blocks, the JPEG algorithm is made more flexible. The boundaries of these polygonal blocks matched the boundaries of objects and allowed the resulting object-orient image compression to achieve higher compression ratio [17].

In [18], a new method is proposed for post-processing of JPEG-encoded images in order to reduce coding artifacts and enhance visual quality. This method simply re-applies JPEG to the shifted versions of the already compressed image. However, the approach does not specifically count the discontinuities in the block boundaries, neither does it make direct use of the smoothness criteria. It uses the JPEG process itself to reduce the compression artifacts of the JPEG encoded image.

On the other hand, the work in [15] presents a computationally efficient framework for color image enhancement in the compressed wavelet domain. It proposes a fast image enhancement framework in the compressed wavelet domain, especially for JPEG2000. The proposed approach introduces enhancements in both global and local contrast and brightness as well as preserving color consistency. In this framework, inverse transform is shown to be unneeded for image enhancement since linear scale factors were directly applied to both scaling and wavelet coefficients in the compressed domain, which resulted in high computational efficiency.

Furthermore, deep neural networks are effectively used in solving lossy image compression problems since the late 1980s [21]. In these methods, the basic autoencoder structure is used, and a binary representation for an image is introduced by quantizing either the bottleneck layer or the corresponding variables. In [16], a method for lossy image compression based on recurrent, convolutional neural networks is proposed, while, in [7], Fuzzy C-mean clustering for priority mapping

has also been used as an adaptive quantization mask in order to improve encoding efficiency of JPEG method by keeping possession of image data. As a result, the blocking artefacts and encoding bit rates were reduced, while the compression efficiency for acceptable image quality was enhanced.

The work in [6] presented a modified JPEG image compression method useful for simulation in industry and biomedical applications by utilizing a region-based variable quantization scheme. It uses three masks that are step, linear, and raised cosine interpolated for controlling the quantization granularity which appear in transitions between regions. Meanwhile, image compression using JPEG algorithm results in an unwanted blocking effect in smooth areas, which is generated due to the coarse quantization of DCT coefficients. Singh proposed a deblocking algorithm for filtering those blocked boundaries by making use of smoothening, detection of blocked edges, and filtering only the difference between the pixels that contain the blocked edge [2]. Finally, Hopkins et al. improved JPEG compression quality through searching for new quantization tables that have the ability to decrease the FSIM (feature Similarity Index Measure) error and increase CR (Compression Ratio) at certain quality levels [22].

In this paper, a new scheme is implemented to enhance the JPEG compression algorithm to have better compression ratios compared to JPEG in the case of large images. The paper is organized as follows; Section 2 explains the methodology for the proposed method where image compression and decompression algorithms are given in detail, and Section 3 provides the experimental results for implementing the algorithm explained in Section 2. Here, many cases and tests are examined. Finally, Section 4 concludes the work done in this paper.

2. Methodology

The proposed scheme in this paper represents a simple, yet powerful technique for image compression where the JPEG algorithm is enhanced to achieve better compression ratio (CR), and higher Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) for large size images. This scheme is described in the following:

2.1. Pre-Processing

In this step, we have:

1. The image size is adjusted to make it divisible to 4×4 blocks. Let R and C be the image width and length, respectively, then R and C are changed to:

$$R_{new} = R - \text{mod}(R, 4)$$

$$C_{new} = C - \text{mod}(C, 4)$$

where R_{new} and C_{new} are divisible by 4, and the image size is adjusted to a $R_{new} \times C_{new}$ blocks.

2. To soften the boundaries of the image, padding is added to the image borders with replicated values of the nearest points.
3. Next, the image is divided into a non-overlapping 4×4 blocks.

2.2. Image Compression

In the image compression, the following procedure is undertaken:

1. The four corner points of each 4×4 non-overlapped blocks of an image are selected.
2. The average value for each edge point with the edge points of neighbor blocks is found as shown in Figure 1. Each 4×4 block is represented by this average value and accordingly a 512×512 image is compressed to a 128×128 image.

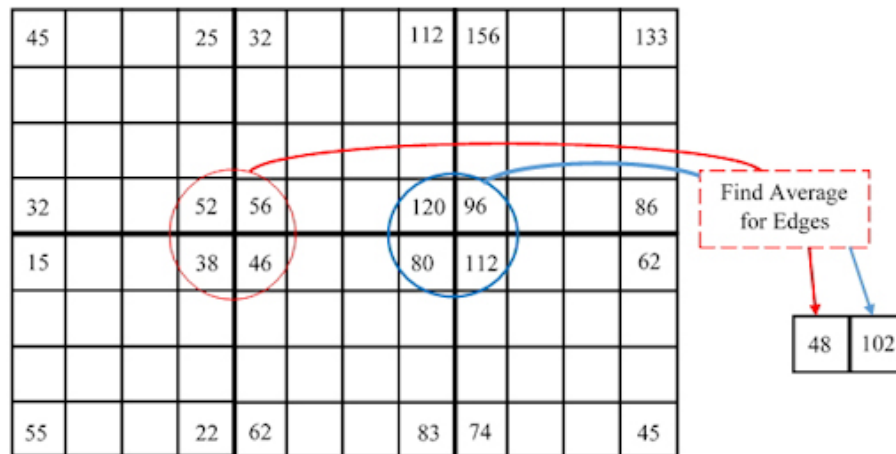


Figure 1. Averaging the values for edge points of neighbor blocks.

3. The JPEG compression method is carried out for the resultant image from the previous step and further compression is performed.
4. The compressed image is stored.

The details of the proposed image compression framework are described in Algorithm 1.

Algorithm 1: Image Compression

Input: Image I of dimensions $R \times C$

Output: Compressed Image W of dimension $R_{new}/4 \times C_{new}/4$

- 1 Preprocessing make the dimensions of the image divisible by 4 ($R_{new} \times C_{new}$), using repeated padding.
 - 2 For each 4×4 non-overlapping block B_{ij}
Read each block and select only the four corner values.
Save these four values to create a new image called F .
 - 3 F is the resultant image of dimensions $R_{new}/2 \times C_{new}/2$
 - 4 For each 2×2 non-overlapping block K_{ij}
Calculate the average of these 4 pixel values.
Save this value (average of pixel values) to create a new image called G .
 - 5 G is the resultant image of dimensions $R_{new}/4 \times C_{new}/4$
 - 6 Save G as JPEG format to get output image called W .
-

2.3. Image Decompression

Using the compressed image, the following steps can be performed to reconstruct the principal content for the original image:

1. The JPEG decompression method is implemented for the compressed image.
2. For each four points, construct a 2×2 matrix, then the tanh function presented in [11] is used to estimate a 2×4 matrix from the resulting image in step 1 as shown in Figure 2b:

$$x(1, j) = a + (b - a) \times (\tanh(2 \times (j - 1)/4)),$$

$$x(4, j) = c + (d - c) \times (\tanh(2 \times (j - 1)/4)).$$
3. For each column of the 2×4 matrix, the tanh function presented in [11] is reimplemented to estimate the other points for constructing the decompressed 4×4 blocks as shown in Figure 2c:

$$x(i, j) = x(1, j) + [x(4, j) - x(1, j)] \times (\tanh(2 \times (i - 1)/4)).$$
4. Let g be the original image, and c be the decompressed image; if $(g - c) \neq 0$, then c is scaled up or down to match g .
5. To determine the quality of the decompressed image, PSNR and SSIM have to be calculated.

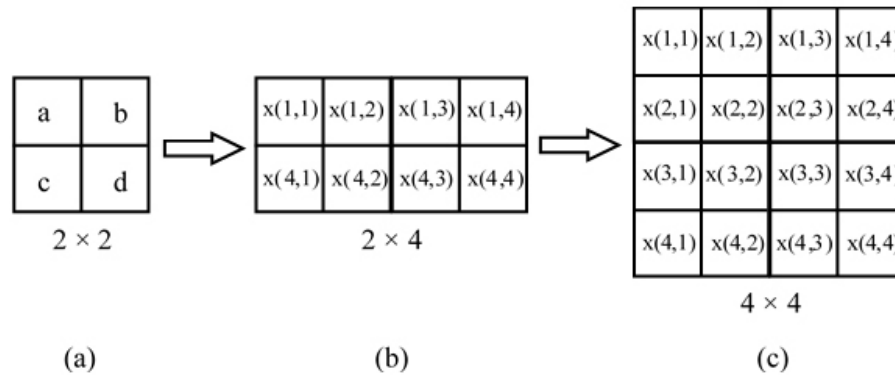


Figure 2. Constructing the decompressed block. (a) 2×2 points block, (b) 2×4 row stretch block, (c) 4×4 reconstructed block.

The details of the proposed image decompression framework are described in Algorithm 2.

Algorithm 2: Image Decompression

- Input:** Image W of dimensions $R_{new}/4 \times C_{new}/4$
Output: Reconstructed image \hat{I} of dimensions $R \times C$
- 1 Convert the image W to a numerical array.
 - 2 For each 2×2 overlapped block.
 - 3 Stretch each row to 1×4 as follows:
 Let a, b, c and d are the input points of the 2×2 block.
for $k_3 = 1$ **to** 5 **do**
 4 $x(1, k_3) = a + (b - a) \times (\tanh(2 \times (k_3 - 1)/4))$;
 5 $x(4, k_3) = c + (d - c) \times (\tanh(2 \times (k_3 - 1)/4))$;
end
 6 **end**
 - 7 Stretch each column to 4×1 as follows:
for $k_1 = 1$ **to** 5 **do**
 8 **for** $k_2 = 1$ **to** 5 **do**
 9 $x(k_1, k_2) = x(1, k_2) + [x(4, k_2) - x(1, k_2)] \times (\tanh(2 \times (k_1 - 1)/4))$;
 10 **end**
 11 **end**
 - 12 The resulted 4×4 block is saved in a new image called T with dimensions $R \times C$.
 - 13 Compare the corners values of each 4×4 overlapped block with the original 2×2 values to remove any expected blocking effect using Algorithm 3.
 - 14 Save the reconstructed image called \hat{I} .
-

Algorithm 3: Blocking Effect Removal

- Input:** Original 2×2 block, reconstructed 4×4 block.
Output: Reconstructed 4×4 corrected block.
- 1 Calculate the correction factor r .
 let $(a, b, c$ and $d)$ are the input points of the original 2×2 block and $(a_1, b_1, c_1$ and $d_1)$ are the corners of the 4×4 reconstructed block.

$$r = \frac{a}{a_1} + \frac{b}{b_1} + \frac{c}{c_1} + \frac{d}{d_1}.$$
 - 2 Multiply all the points of the 4×4 block by $r/4$.
 - 3 Save the result as 4×4 corrected block.
-

2.4. Quality Analysis of the Proposed Approach

In general, the proposed approach can be used to compress images of large size and high quality. It can reduce the size of the image to 1/16 of the original size. Then, apply the JPEG algorithm. As a result, the proposed approach reduces the total size up to 15%–25% of the size of the JPEG for the same quality. The proposed approach also minimizes the total number of mathematical operations (number of multiplications) by up to 10% on the compression side and by up to 20% on the decompression side.

The disadvantage of the proposed approach is the additional error (offset error as mentioned in test 1). This error is considered as a fixed value for the image at any CR. However, the produced error in the proposed approach could be increased slowly with CR (see Test 4). Therefore, the proposed approach becomes efficient for high CR values, but it is not suitable for low CR values. As a result, we need to add a simple optimization block in the compression system to switch between the traditional JPEG method and the proposed method at a suitable point to increase the quality of the reconstructed image.

3. Experimental Results

The proposed scheme explained in Section 2 has been conducted on gray scale images, 2 sized (512×512), one sized (1024×1024), and three sized (1920×1080). These images are shown in Figure 3a–f, respectively.

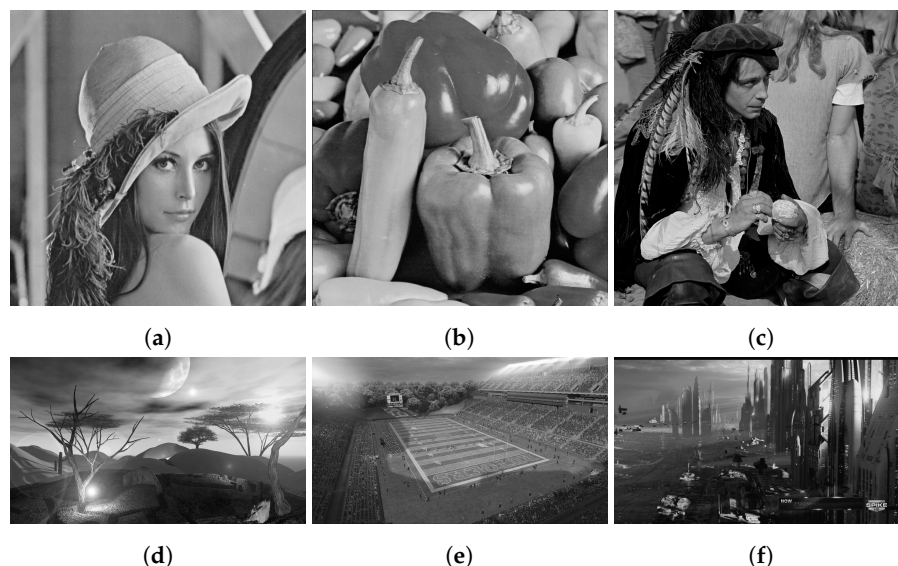


Figure 3. Images used in this paper work with their sizes as follows: (a) (512×512); (b) (512×512); (c) (1024×1024); (d) (1920×1080); (e) (1920×1080); (f) (1920×1080).

To provide objective judgment of the proposed method, two major quality measurements are used. These measurements are the compression ratio (CR), defined as dividing the file size of the original uncompressed image by that of the compressed image [8].

The other measurement is the peak signal to noise ratio (PSNR) given as [8,23]:

$$PSNR = 10 \log_{10} \frac{(255^2 XY)}{\sum_x \sum_y (g(x, y) - \hat{g}(x, y))^2}, \quad (1)$$

where g and \hat{g} are the original and reconstructed image pixel value, respectively, $x = 1, \dots, X$ and $y = 1, \dots, Y$, where X and Y are image dimensions. In addition, Structural Similarity Index (SSIM) is used as a subjective quality measurement for the test images besides the PSNR, SSIM value ranges between 0.0–1.0, where low value means large structural variation, and vice versa [11,22].

Four tests are carried out to show the improvements of the proposed method compared to JPEG compression. In tests (2–4), the standard JPEG algorithm and the proposed compression method are

applied to the images given in Figure 3, and different simulations are run. The simulation results for these tests are given in Tables 1–3, respectively. Three measurements are considered in each test, these measurements are the image size, PSNR, and CR, for both JPEG and the proposed algorithms. The four tests are organized as:

3.1. Test 1: Tanh Function Effect

This test shows the performance of the proposed compression method; however, its procedure steps are done without the JPEG compression part to show the advantage of using the smooth filter with the tanh function to enhance the quality of the constructed image. In this process, noise is added to the original image. Generally, the obtained CR equals 16, while PSNR value is between 28.5–34.5 and SSIM value is between 0.89–0.97. This illustrates that the proposed steps have good PSNR values with improved CR.

3.2. Test 2: Fixing PSNR

In this test, the standard JPEG method and the proposed method are applied to the images shown in Figure 3, and the simulation results are given in Table 1. Here, PSNR values for the six images are adjusted to approximately the same value for the two algorithms, CR and SSIM values are measured and compared.

Table 1. Results of Test 2.

	Image (a)	Image (b)	Image (c)	Image (d)	Image (e)	Image (f)
Original size	257 K ¹	257 K	1 M ²	1.97 M	1.97 M	1.97 M
PSNR of JPEG method	28.86	26.95	26.52	29.04	26.95	28.69
PSNR of proposed method	28.84	27.39	26.92	29.13	27.00	28.70
SSIM of JPEG method	0.7957	0.7455	0.6783	0.8135	0.6932	0.8232
SSIM of proposed method	0.8246	0.8200	0.7241	0.8371	0.6771	0.8411
Size using JPEG method	4.87 k	4.99 k	18.6 k	31.5 k	30.6 k	30.5 k
Size using proposed method	2.41 k	2.91 k	10.2 k	10.9 k	5.49 k	5.88 k
CR using JPEG method	53	52	54	63	66	66
CR using proposed method	107	88	98	181	367	343

¹ Kilo Byte, ² Mega Byte.

We observe an improvement in CR in terms of using the proposed algorithm by factor more than 1 (from 1.7 for low quality images to 5.6 for high quality images). Furthermore, the quality of the images is enhanced using the proposed method over JPEG method as listed in Table 1.

3.3. Test 3: Fixing the Size of the Images

In this test, the standard JPEG method and the proposed method are applied to the images shown in Figure 3 and the simulations results are given in Table 2 and are shown in Figure 4. Here, the sizes for the six images are adjusted to approximately the same value for the two algorithms; PSNR and SSIM values are measured and compared.

Table 2. Results of Test 3.

	Image (a)	Image (b)	Image (c)	Image (d)	Image (e)	Image (f)
Original size	257 K	257 K	1 M	1.97 M	1.97 M	1.97 M
PSNR of JPEG	25.66	24.93	24.65	26.14	24.93	26.66
PSNR of proposed method	29.71	28.15	27.59	30.15	29.60	33.19
SSIM of JPEG	0.7228	0.6923	0.5911	0.7708	0.6138	0.7851
SSIM of proposed method	0.8588	0.8529	0.7638	0.8722	0.7789	0.9301
Size using JPEG	3.93 k	4.27 k	15.4 k	26.8 k	26.8 k	27.7 k
Size using proposed	3.93 k	4.23 k	15.4 k	26.7 k	26.6 k	27.8 k
CR using JPEG	76	60	65	73	73	71
CR using proposed	76	60	65	74	74	71

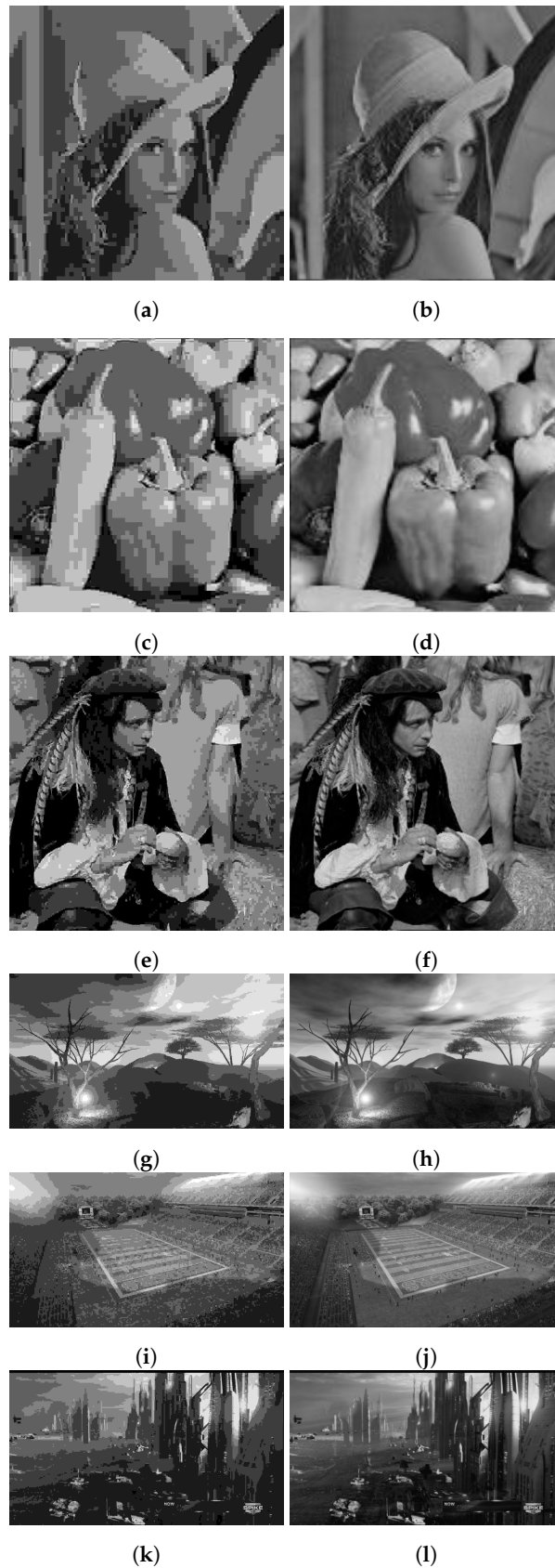


Figure 4. Simulation Results of test 3, left-hand column (a,c,e,g,i,k): results of standard JPEG method, right-hand column (b,d,f,h,j,l): results of our proposed method.

We observe an improvement in PSNR in terms of the proposed algorithm by 3 dB to 4 dB and 0.1 to 0.17 in SSIM value for the same size. Furthermore, the quality of the images is enhanced using the proposed method over JPEG method as listed in Table 2.

3.4. Test 4

The standard JPEG method and the proposed method are applied to the high quality image shown in Figure 3d to show the advantages of the proposed method in this test, and the simulation results are given in Table 3.

Table 3. Results of Test 3.

Image (d)	Simulation A	Simulation B
Original size	1.97 M	1.97 M
PSNR of JPEG	26.45	26.14
PSNR of proposed method	30.17	28.15
SSIM of JPEG	0.7797	0.7708
SSIM of proposed method	0.8736	0.8147
Size using JPEG	27.8 k	26.6 k
Size using proposed	28.1 k	6.47 k
CR using JPEG	71	74
CR using proposed	70	304

Here, two simulations are considered, these are:

- A When Q for the proposed method is high (=88), then the proposed method is +3.7 dB higher than JPEG with the same CR value.
- B When Q for the proposed method is low (=20), then the proposed method is +2 dB higher than JPEG and the CR value for the proposed method is more than four times that for JPEG.

Table 3 shows these two simulations, showing that the proposed method is more efficient for high CR and high quality images. Generally, the standard JPEG has CR < 64 (with PSNR = 29 as in the image in Figure 3d in Test-2 while the proposed method has CR < 300 (with PSNR = 28 as in the image in Figure 3d in Test-4).

The curves shown in Figure 5 represent the relationship between the PSNR and CR for JPEG and the proposed algorithm measured for the image given in Figure 3d and used in Test-4. The curves show that, for this large image, the new approach achieves better compression ratio than ordinary JPEG. Figure 6 shows the result at CR (=74) with magnification for the original image, image of JPEG method, and our method image, respectively. It is obvious that the features of the reconstructed image using our proposed method show less blocking effects and blurring artifacts in comparison with the JPEG method for the same image size.

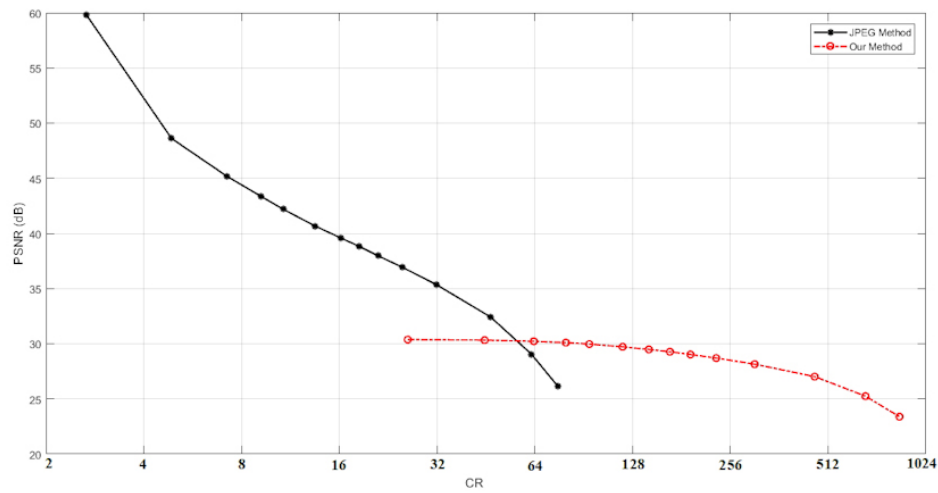


Figure 5. PSNR vs. CR for JPEG only and the proposed algorithm.

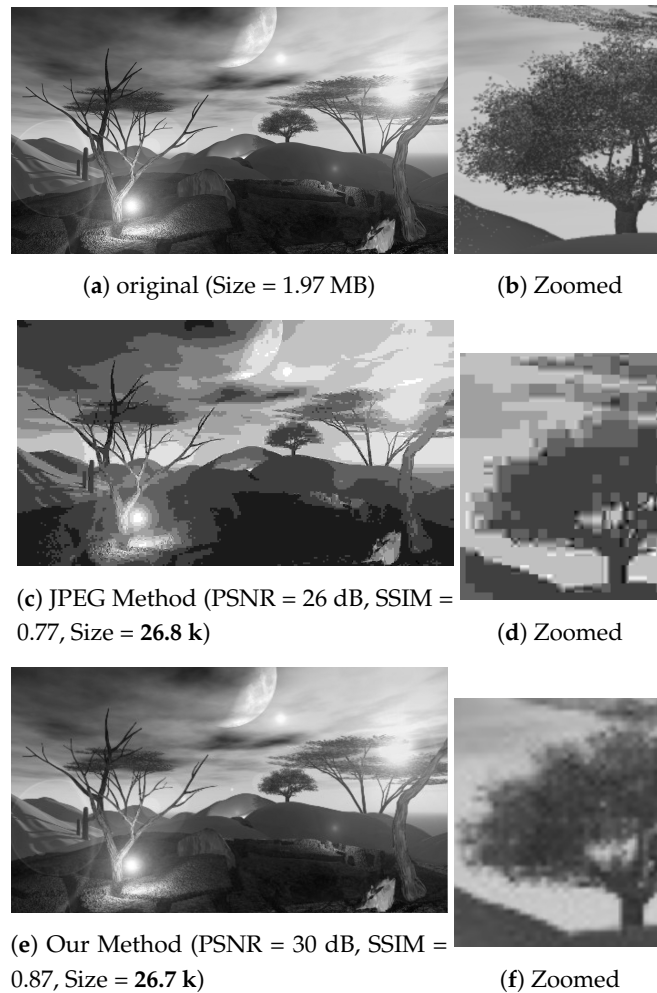


Figure 6. Simulation results for the same CR (=74), and the image magnification for the same part. (a) original Image, (b) zoom of selected of a, (c) JPEG method result, (d) zoom of selected part of c, (e) our method result, (f) zoom of selected part of e.

4. Conclusions

This paper presents a novel approach which enables the JPEG method to compress large monochrome images. This approach improves the PSNR, SSIM, and CR values for the compressed images. Furthermore, for high quality images, it can provide very high CR value as illustrated in Test-4. The proposed scheme uses a smooth filter with the hyperbolic (tanh) function to enhance the quality of the reconstructed image. The proposed method is considered as a stand-alone compression method as shown in Test-1, where CR equals 16. Adding the proposed method to JPEG improved the overall CR value. Furthermore, it improves the edges of the reconstructed images over the standard JPEG approach.

The experimental results show that better performance can be achieved in terms of PSNR, SSIM, CR and visual quality using the proposed method. The CR limit of JPEG is up to 100 times while the limit of the proposed method is higher than 1000 times. For future work, our proposed method could be applied with other image or data compression methods by substituting the JPEG approach with other compression approaches.

Author Contributions: Conceptualization, D.Z. and W.K.; methodology, D.Z.; software, W.K.; validation, A.A.G. and W.K.; formal analysis, D.Z.; investigation, A.A.G.; resources, A.A.G.; data curation, W.K.; writing—original draft preparation, A.A.G.; writing—review and editing, W.K.; visualization, D.Z.; supervision, D.Z.

Funding: This research received no external funding.

Acknowledgments: We would like to present our thanks to Mustansiriyah university for supporting our experiments in providing us all the necessary data and software.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CR	Compression Ratio
PSNR	Peak Signal-to-Noise Ratio
bpp	bits per pixel
Q	Image Quality
SSIM	Structural Similarity Index
JPEG	Joint Photographic Experts Group

References

1. Hussain, A.J.; Al-Fayadh, A.; Radi, N. Image Compression Techniques: A Survey in Lossless and Lossy algorithms. *Neurocomputing* **2018**, *300*, 44–69. [\[CrossRef\]](#)
2. Singh, S. An Algorithm For Improving The Quality Of Compacted JPEG Image By Minimizes The Blocking Artifacts. *Int. J. Comput. Graph. Animat.* **2012**, *2*, 17–35. [\[CrossRef\]](#)
3. Li, H.; Wen-yan, W. Improved Method to Compress JPEG Based on Patent. In Proceedings of the International Conference on Educational and Network Technology, Qinhuangdao, China, 25–27 June 2010; pp. 159–162. [\[CrossRef\]](#)
4. Dorobantiu, A.; Brad, R. Improving Lossless Image Compression with Contextual Memory. *Appl. Sci.* **2019**, *9*, 2681. [\[CrossRef\]](#)
5. Hu, J.; Deng, J.; Wu, J. Image Compression Based on Improved FFT Algorithm. *J. Netw.* **2011**, *6*, 1041–1048. [\[CrossRef\]](#)
6. Golner, M.; Mikhael, W.; Krishnang, V. Modified jpeg image compression with region-dependent quantization. *Circuits Syst. Signal Process.* **2002**, *21*, 163–180. [\[CrossRef\]](#)
7. Sombutkaew, R.; Chitsobhuk, O.; Prapruttam, D.; Ruangchaijatuporn, T. Adaptive quantization via fuzzy classified priority mapping for liver ultrasound compression. *Int. J. Innov. Comput. Inf. Control* **2016**, *12*, 635–649.
8. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2006; ISBN 013168728X.

9. Hassan, S.A.; Hussain, M. Spatial domain lossless image data compression method. In Proceedings of the International Conference on Information and Communication Technologies, Karachi, Pakistan, 23–24 July 2011; pp. 1–4. [\[CrossRef\]](#)
10. Sajikumar, S.; Anilkumar, A.K. Image compression using chebyshev polynomial surface fit. *Int. J. Pure Appl. Math. Sci.* **2017**, *10*, 15–27.
11. Khalaf, W.; Zaghar, D.; Hashim, N. Enhancement of Curve-Fitting Image Compression Using Hyperbolic Function. *Symmetry* **2019**, *11*, 291. [\[CrossRef\]](#)
12. Cabeen, K.; Gent, P. Image Compression and the Discrete Cosine Transform. In *Math 45*; College of the Redwoods: Eureka, CA, USA, 1998; pp. 1–11.
13. Dagher, I.; Saliba, M.; Farah, R. Combined DCT-Haar Transforms for Image Compression. In Proceedings of the 4th World Congress on World Congress on Electrical Engineering and Computer Systems and Science, Madrid, Spain, 21–23 August 2018; pp. 1–8. [\[CrossRef\]](#)
14. Doukas, C.N.; Maglogiannis, I.; Kormentzas, G. Medical Image Compression using Wavelet Transform on Mobile Devices with ROI coding support. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 1–4 September 2005; pp. 3779–3784.
15. Cho, D.; Bui, T.D. Fast image enhancement in compressed wavelet domain. *Signal Process.* **2014**, *98*, 295–307. [\[CrossRef\]](#)
16. Johnston, N.; Vincent, D.; Minnen, D.; Covell, M.; Singh, S.; Chinen, T.; Hwang, S.; Shor, J.; Toderici, G. Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4385–4393.
17. Ding, J.; Huang, Y.; Lin, P.; Pei, S.; Chen, H.; Wang, Y. Two-Dimensional Orthogonal DCT Expansion in Trapezoid and Triangular Blocks and Modified JPEG Image Compression. *IEEE Trans. Image Process.* **2013**, *22*, 3664–3675. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Nosratinia, A. Enhancement of JPEG-Compressed Images by Re-application of JPEG. *J. VLSI Signal Process.* **2001**, *27*, 69–79. [\[CrossRef\]](#)
19. Kacem, H.L.H.; Kammoun, F.; Bouhlel, M.S. Improvement of The Compression JPEG Quality by a Pre-processing Algorithm Based on Denoising. In Proceedings of the 2004 IEEE International Conference on Industrial Technology, Hammamet, Tunisia, 8–10 December 2004; pp. 1319–1324. [\[CrossRef\]](#)
20. Kohno, K.; Tanaka, A.; Imai, H. A novel criterion for quality improvement of JPEG images based on image database and re-application of JPEG. In Proceedings of the 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, Hollywood, CA, USA, 3–6 December 2012; pp. 1–4.
21. Cottrell, G.W.; Munro, P.; Zipser, D. Image Compression by Back Propagation: An Example of Extensional Programing. In *Advances in Cognitive Science*, 2nd ed.; Institute for Cognitive Science, University of California: San Diego, CA, USA, 1987; pp. 208–240.
22. Hopkins, M.; Mitzenmacher, M.; Wagner-Carena, S. Simulated annealing for jpeg quantization. *arXiv* **2017**, arXiv:1709.00649.
23. Chiranjeevi, K.; Jena, U.R. Image compression based on vector quantization using cuckoo search optimization technique. *Ain Shams Eng. J.* **2018**, *9*, 1417–1431. [\[CrossRef\]](#)



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).