Image Compression based on Quadtree and Polynomial

Ghadah Al-Khafaji, Ph.D
Dept. of Computer Science,
Baghdad University,
College of Science.

ABSTRACT

In this paper, an efficient image compression scheme is introduced, it is based on partitioning the image into blocks of variable sizes according to its locally changing image characteristics and then using the polynomial approximation to decompose image signal with less compressed information required compared to traditional predictive coding techniques, finally Huffman coding utilized to improve compression performance rate. The test results indicate that the suggested method can lead to promising performance due to simplicity and efficiency in terms of overcoming the limitations of predictive coding and fixed block size.

General Terms

Quadtree partitioning of variable block sizes with polynomial approximation for lossy image compression.

Keywords

Image compression, compression techniques, quadtree and polynomial representation.

1. INTRODUCTION

Image compression techniques generally fall into two categories: lossless and lossy depending on the redundancy type exploited, where lossless also called information preserving or error free techniques, in which the image compressed without losing information that rearrange or reorder the image content, and are based on the utilization of statistical redundancy alone such as Huffman coding, Arithmetic coding and Lempel-Ziv algorithm, while lossy which remove content from the image, which degrades the compressed image quality, and are based on the utilization of psycho-visual redundancy, either solely or combined with statistical redundancy such as such as vector quantization, fractal, transform coding and JPEG, reviews of lossless and lossy techniques can be found in [1]-[8].

In general, lossy techniques work on segment based that subdivide the image into non-overlapping segments (blocks) of fixed sizes or variable sizes. Typically the fixed partitioning method is adopted due to its simplicity and popularity, but this is at the expense of efficiency, and comes with a greater storage cost, because the blocks are partitioned based on the size of the region, regardless of the content, whether that region or block is uniform or non-uniform [9]. The variable block partitioning methods utilized by a number of researchers [10]-[19], to overcome the fixed partitioning method drawback using partitioning techniques such as quadtree, HV (horizontal-vertical) and triangular, in which the results are promising but still under development, and not yet a recognized to be used by standard techniques due to complexity or difficulty of choosing the uniformity measure and time required.

Nowadays, there's increase trend of utilizing the polynomial approximation representation [20]-[21] due to its simplicity, symmetry of encoder and decoder and high compression rates can provide where no need to extra information to be used like seed values compared to the traditional predictive coding method [22]-[32].

In this paper, a lossy quadtree variable block partitioning method along with the polynomial approximation is introduced to remove the redundancy between neighboring pixels according to its local dependency that efficiently improve the quality and the compression rate. The rest of the paper organized as follows, section 2 contains comprehensive clarification of the proposed system; the results of the proposed system, is given in section 3.

2. THE PROPOSED SYSTEM

The main taken concerns in the proposed system are:

- Get the benefit of hierarchical partitioning representation where blocks of variable sizes produced that efficiently improve the compression rates and quality.
- Since in this paper the linear polynomial representation is adopted to remove the spatial redundancy, the coefficients are fixed within the subdivided block-by-block image, so three coefficients (a_0,a_1,a_2) are required to represent each block. Therefore, the performance vary according to the blocks nature, in other words the performance increase when applied to large smooth regions and reduced when applied to edge regions.
- The entropy coding using Huffman techniques used efficiently in order to minimize the bit required.

The implementation of the proposed system is explained in the following steps, the layout of the encoder is illustrated in Figure 1:

Step 1: Load the input uncompressed image I of size $N\times N$
Step 2: Partition the image I into non-overlapped blocks of variable sizes $n\times m$ using a quadtree partitioning scheme by checking the uniformity of the tested block, that start with partitioning (dividing) the image into blocks whose size is equal to the maximum block size, if the block is not uniform then the partitioning repeated on its four quadrants in hierarchical manner until reaching the minimum block size where the uniformity condition is satisfied.
The uniformity criteria based implicitly on utilizing the mean and standard deviation of quad region, where for non-uniform region it exceeds a certain Standard Deviation Threshold Value and their mean intensity, is limited between Minimum Mean Threshold Value and Maximum Mean Threshold Value. After that the constructed quadtree will consists of partitions whose size value will be between minimum and maximum block size. Algorithm 1 summarizes the quadtree partitioning steps.

Step 3: Perform the polynomial representation to the variable blocks sizes that resultant of step2 according to equations (1,2,3) [20].

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

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

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

Where *I(i,j)* is the original image block of size $n \times m$ and

$$x_c = \frac{n-1}{2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$$y_c = \frac{m-1}{2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Step 4: Apply uniform scalar quantization to quantize the polynomial approximation coefficients, where each coefficient is quantized using different quantization step.

$$a_0 Q = round(\frac{a_0}{QS_{a0}}) \to a_0 D = a_0 Q \times QS_{a0} \ldots\ldots (6)$$

$$a_1 Q = round(\frac{a_1}{QS_{a1}}) \to a_1 D = a_1 Q \times QS_{a1} \ldots\ldots (7)$$

$$a_2 Q = round(\frac{a_2}{QS_{a2}}) \to a_2 D = a_2 Q \times QS_{a2} \ldots\ldots (8)$$

Where a_0Q, a_1Q, a_2Q are the polynomial quantized values, $QS_{a0}, QS_{a1}, QS_{a2}$ are the quantization steps of the polynomial coefficients, and a_0D, a_1D, a_2D are polynomial dequantized values.

Step 5: Determine the predicted or approximated image value $\tilde{I}$ using the dequantized polynomial coefficients for each encoded block representation:

$$\tilde{I} = a_0 D + a_1 D(j - x_c) + a_2 D(i - y_c) \ldots\ldots\ldots\ldots\ldots (9)$$

Step 6: Find the residual or prediction error as difference between the original *I* and the predicted one $\tilde{I}$.

$$R(i, j) = I(i, j) - \tilde{I}(i, j) \ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Step 7: Perform scalar uniform quantization to quantize the residual part, where residual value is divided by the quantization step. The quantization step values affected the image quality and the compression rate.

Step 8: Apply Huffman coding techniques to remove the rest of redundancy that embedded between the quantized values of the residual and the polynomial coefficients.

To reconstruct the decompressed image all the above mentioned steps are reversed as shown in Figure 2.

3. EXPERIMENTS AND RESULTS

For testing the proposed system performance; it is applied on a number of well-known standard images (see Figure 3 for an overview), all images of 256 gray levels (8bits/pixel) of size 256×256.

The tests have been performed using variable block sizes of different Minimum &Maximum block sizes and compare it with fixed block size {4×4}, the quantization levels utilized was selected to be between 4 and 64 levels, using 2 to 6 bits on both the residual image and the approximation representation coefficients (a_0,a_1,a_2). The partitioned images for fixed and quadtree schemes are shown in Figures 4, 5, 6 and 7 respectively.

The compression ratio, which is the ratio of the original image size to the compressed size along with the normalized root mean square error (NRMSE) between the original image I and the decoded image $\hat{I}$ was adopted as a fidelity or degradation measure as in equation (11), where the range of the values is between 0 and 1. A value near zero indicates high image quality, i.e. the decoded image closely resembles the original, and vice versa.

$$NRMSE(I, \hat{I}) = \sqrt{\frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [\hat{I}(x, y) - I(x, y)]^2}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y)^2}} \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Certainly, the quality of the decoded image is improves as the number of quantization levels of both the approximation representation coefficients and residual image increase. The main disadvantage of increasing the quantization levels, however, lies in increasing the size of the compressed information. It is a trade-off between the desired quality and the consumption of bytes; the higher the quality required, the larger the number of quantization levels that must be used.

The results shown in Table 1 and Figure 8 illustrates that the 4×4 fixed block size, compared to quadtree of nearly same number of blocks as fixed block case, of the three tested images. The results show that for quadtree higher quality produced due to utilizing small blocks of finer details, but with less compression rate where the using of bigger block sizes implicitly meaning a decrease in modelling fidelity as the block gets bigger where the size of residual varies according to the block size (i.e., residual size increase due to insufficient model flexibility). Also the result demonstrates that the compression rates is directly affected by the residual size (residual burden) not the size of polynomial approximation coefficients, in other words even with less coefficient parameters required in quadtree the residual represents the exhausted bytes.

Table 2, shown the high quality results for the three tested images, using quadtree partitioning method of different block sizes. The results clearly show that the quality improves with the increase of the number of partitioned blocks, and vice versa.

The decoded images with different quality status are shown in Figures 9,10 and 11 respectively.

Lastly, the results showed that the quality of the decoded image dose not suffers from the blocking effects and edge degradation as the block gets bigger, which differs from other compression techniques, that assumed that as the block got smaller, higher quality would be achieved. The main reason of the improving of image quality in the polynomial approximation coding techniques using bigger block sizes due to dominating residual image.

4. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

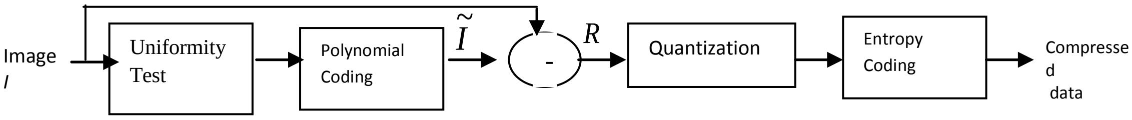


Fig 1. Encoder structure of the proposed system.

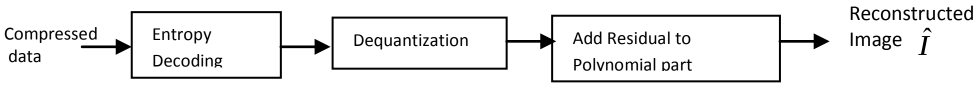

Compressed data
Entropy Decoding
Dequantization
Add Residual to Polynomial part
Reconstructed Image $\hat{I}$

Fig 2. Decoder structure of the proposed system.

Algorithm 1. Quadtree partitioning algorithm.

```
1. Input image $I$ of size $N×N$
2. Select the Minimum and Maximum Block Size
3. Test the Uniformity criteria
   a) If (region <= Minimum Block Size) then uniform
   b) Else if (region > Maximum Block Size) then nonuniform
   c) Else
   If (region > Standard Deviation Threshold Value) and (region > Minimum Mean Threshold Value
   and region < Maximum Mean Threshold Value) then nonuniform
   Else uniform
```

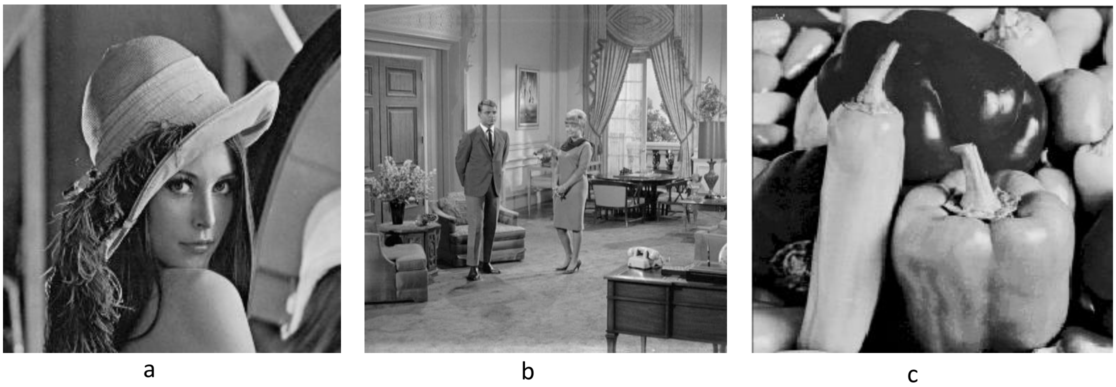


Fig 3. Overview of the tested images (a) Lena image, (b) Living room image and (c) Paper image, all images of size 256×256, gray scale images.

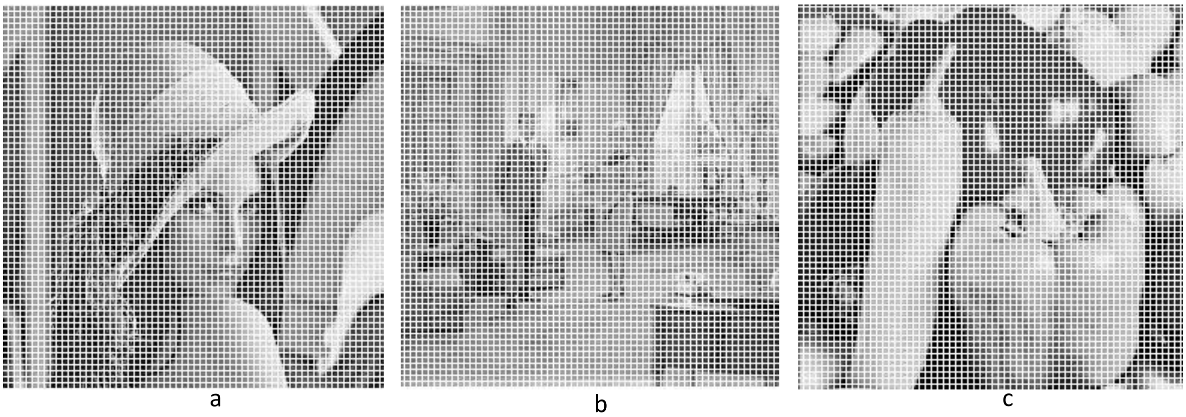


Fig 4. Fixed Partitioning of block size 4×4 on the tested images of sizes 256×256 (a) Lena image, (b) Living room image and (c) Paper image, number of blocks 4096 in all images.

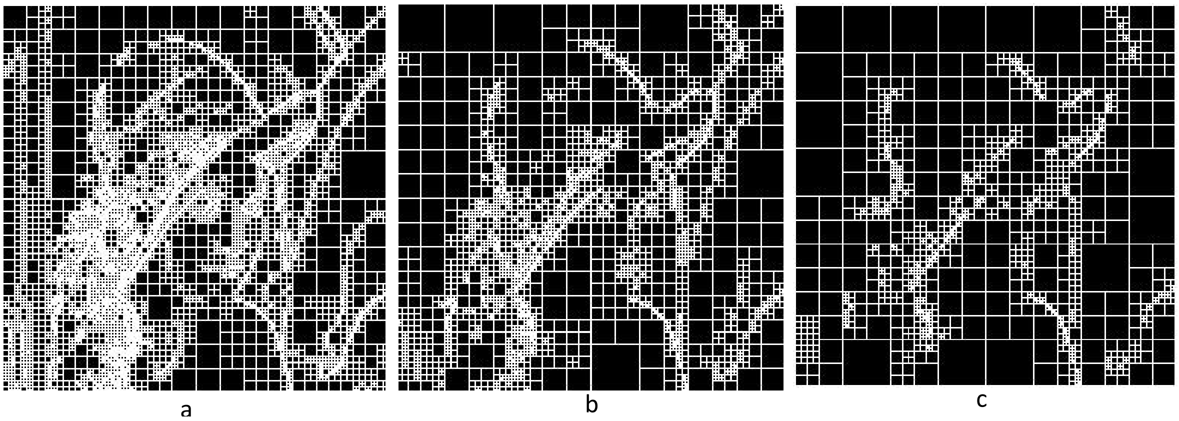


Fig 5(a-c). Quadtree partitioning applied on Lena image with different number of blocks.

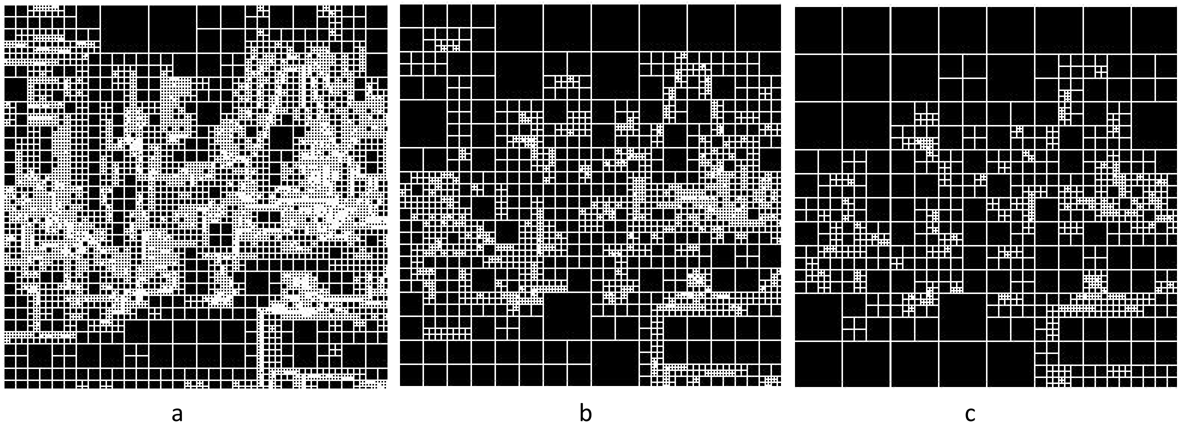


Fig 6(a-c). Quadtree partitioning applied on Living room image with different number of blocks.

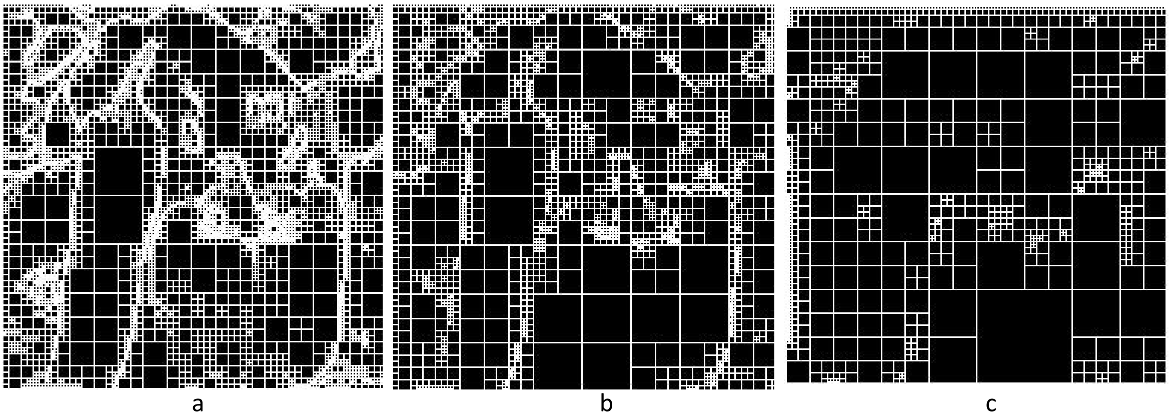


Fig 7(a-c). Quadtree partitioning applied on Paper with different number of blocks.

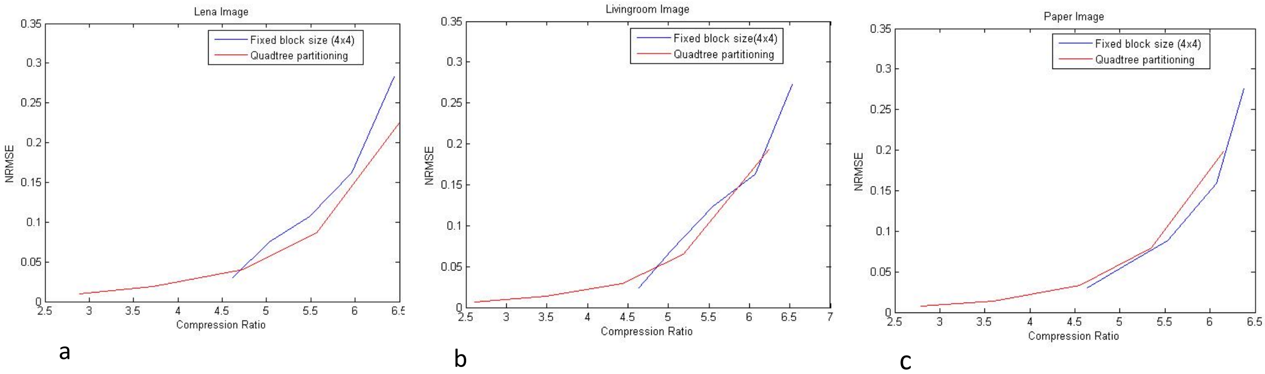


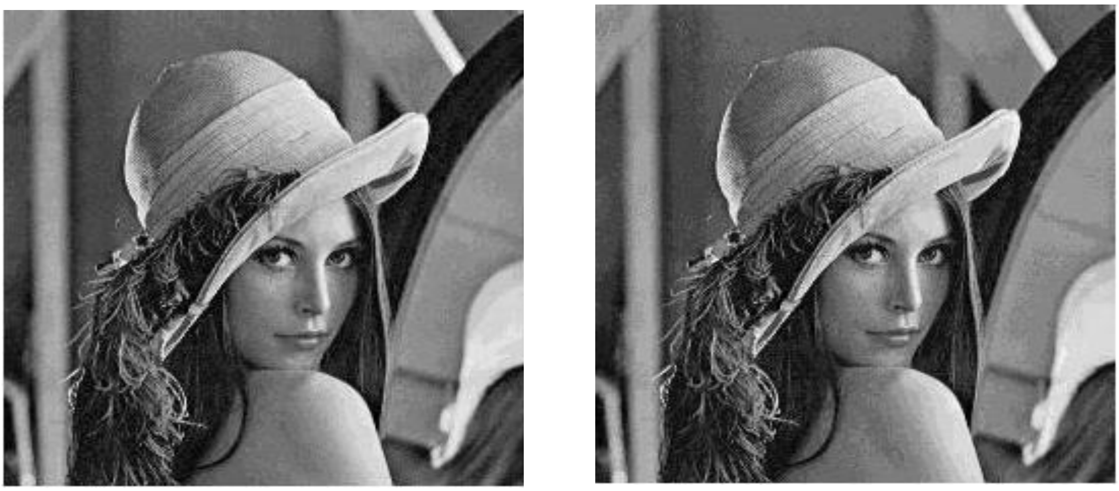
Fig 8. Compression ratio versus the normalized mean square error of the tested images (a) Lena (b)Living room (d) Paper using fixed block and quadtree techniques.

Table 1. Comparison between fixed and quadtree techniques on the tested images.

Tested images	Quant coeff. & Res	Quadtree {Minimum Block Size=2, Maximum Block Size=16, Standard Deviation Threshold Value=10 , Minimum Mean Threshold Value =0 and Maximum Mean Threshold Value =110 }			Fixed block 4×4 No. Blocks=4096	
		N. Blocks	CR	NRMSE	CR	NRMSE
Lena	4 levels	3949	6.4926	0.2245	6.4415	0.2826
	8		5.5709	0.0868	5.9632	0.1622
	16		4.7414	0.0402	5.4823	0.1073
	32		3.7186	0.0187	5.0319	0.0748
	64		2.8932	0.0094	4.6185	0.0305
livingroom	4 levels	4068	6.2416	0.1928	6.5314	0.2728
	8		5.1807	0.0650	6.0693	0.1628
	16		4.4259	0.0287	5.5473	0.1234
	32		3.4960	0.0139	5.0335	0.0686
	64		2.6050	0.0064	4.6315	0.0243
Paper	4 levels	4033	6.1478	0.1977	6.3776	0.2757
	8		5.3446	0.0798	6.0726	0.1595
	16		4.5517	0.0334	5.5267	0.0889
	32		3.6033	0.0138	5.0412	0.0573
	64		2.7869	0.0080	4.6335	0.0297

Tested images	Quadtree partitioning Quant coeff. =4 & Res=32 levels		
	N. Blocks	CR	NRMSE
Lena	10777	2.6756	0.0173
	8413	2.8064	0.0187
	6238	2.9275	0.0198
	4618	3.1042	0.0215
	3592	3.1502	0.0217
livingroom	10861	2.8163	0.0155
	8554	2.9843	0.0162
	6712	3.1448	0.0177
	4022	3.2159	0.0198
	3212	3.2871	0.0200
Paper	10984	2.6200	0.0140
	8875	2.6773	0.0156
	6736	2.8521	0.0180
	4954	2.9579	0.0168
	3361	3.0301	0.0217

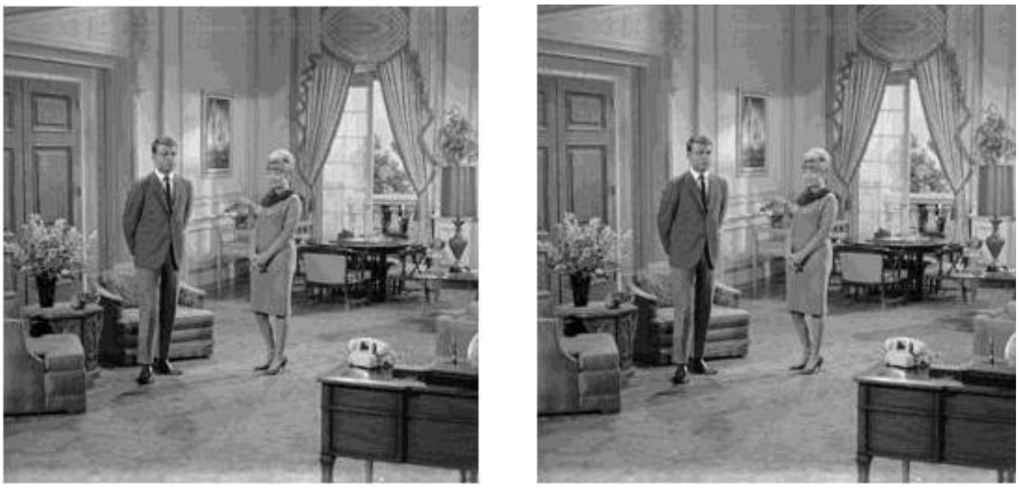
Table 2. Compression performance of the proposed system on the tested images of high quality.



a
No. Blocks=10188
CR=3.361
NRMSE=0.0351

b
No. Blocks=3262
CR=5.412
NRMSE=0.0454

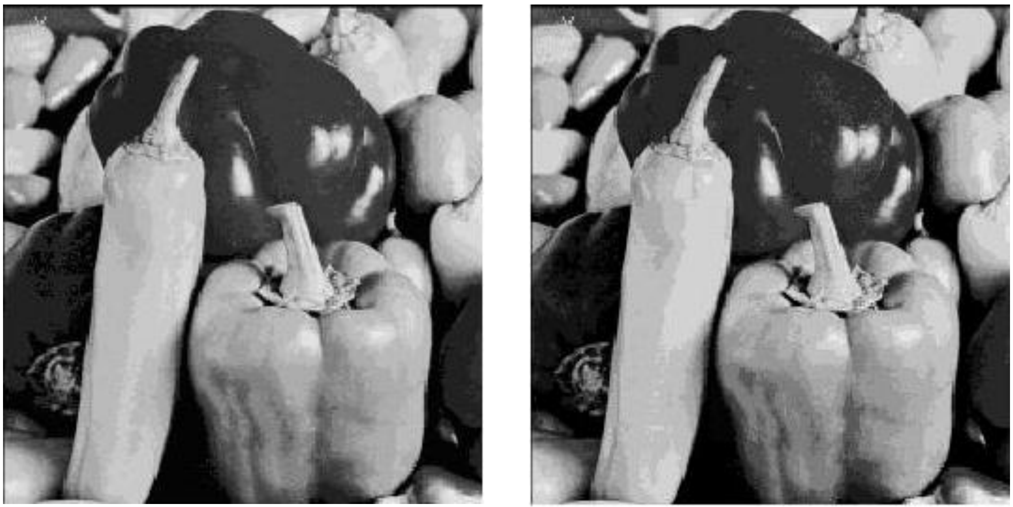
Fig 9. Decompressed Lena image with different quality values using Quant. (coeff. =4 & Res.=16) levels.



a
No. Blocks=9025
CR=3.616
NRMSE=0.0330

b
No. Blocks=6268
CR=4.2867
NRMSE=0.0326

Fig 10. Decompressed Living room image with different quality values using Quant.(coeff. =4 & Res.=16) levels.



a
No. Blocks=9322
CR=3.373
NRMSE=0.0287

b
No. Blocks=3018
CR=5.781
NRMSE=0.0379

Fig11: Decompressed Paper image with different quality values using Quant (coeff. =4 & Res.=16) levels.

5. REFERENCES

[1] Furht, B. 1995. A Survey of Multimedia Compression Techniques and Standards. Real-Time Imaging, 1, 49-67.

[2] Singh, S. K. and Kumar, S. 2010. Mathematical Transforms and Image Compression: A Review. Maejo International Journal of Science and Technology, 4(02), 235-249.

[3] Sachin, D. 2011. A Review of Image Compression and Comparison of its Algorithms. International Journal on Electronics & Communication Technology (IJECT). 2(1), 22-26.

[4] Anitha, S. 2011. 2D Image Compression Technique-A Survey. International Journal of Scientific & Engineering Research, 2(7), 1-6.

[5] Sridevi, S., Vijayakuymar, V.R. and Anuja, R. 2012. A Survey on Various Compression Methods for Medical Images. International Journal of Intelligent Systems and Applications, 3, 13-19.

[6] Vrindavanam, J., Chandran, S. and Mahanti, G. K. 2012. A Survey of Image Compression Methods. Proceedings on International Conference and Workshop on Emerging Trends in Technology 12-17.

[7] Asolkar, P. S., Zope, P. H. and Suralkar S. R. 2013. Review of Data Compression and Different Techniques of Data Compression. International Journal of Engineering Research & Technology (IJERT), 2(1), 1-8.

[8] Amruta, S.G. and Sanjay L.N. 2013. A Review on Lossy to Lossless Image Coding. International Journal of Computer Applications (IJCA), 67(17), 9-16.

[9] Fisher, Y. 1994. Fractal Image Compression: Theory and Application. Springier Verlage, New York.

[10] Vaisey, D. and Gersho, A. 1987. Variable Block-Size Image Coding. . Proceedings of the IEEE international conference on Acoustics, Speech, and Signal Processing, 1051 – 1054.

[11] Wu, P. and Zheng, B. 1998. A New Image Compression Method Based on HV Fractal and DCT. Communication Technology Proceedings, International Conference on ICCT '98. 1, 1-4.

[12] Guorui, J., Yuzhuo, Z., Shiqiang, Y. and Bo, Y. 1999. Fast Fractal Image Compression Based on HV Partition. Part of the SPIE Conference on Multimedia Storage and Archiving Systems. 3846, 473-481.

[13] Jamila, H.S. 2001. Fractal Image Compression , Ph.D. Thesis, College of Science, University of Baghdad.

[14] Ghada, K. T. 2001. Adaptive Fractal Image Compression. M.Sc. Thesis, National Computer Center/Higher Education Institute of Computer and Information.

[15] Golchin, F. and Paliwal, K.K. 2003. Quadtree-based classification in subband image coding. Digital Signal Processing, 13, 656–668.

[16] Rajkumar, W. S., Kulkarni, M.V., Dhore, M.L., Mali, S. N. 2006. Fractal Image Compression Performance Synthesis Through HV Partitioning. Advanced Computing and Communications, ADCOM International Conference on 636 – 637.

[17] Ghada, K.T. and Luay, K. A. 2007. Merge Operation Effect On Image Compression Using Fractal Technique. Journal of Baghdad for Science, 4, 169-173.

[18] Keissarian1, F. 2009. A New Quadtree-based Image Compression Technique using Pattern Matching Algorithm. International Conference on Computational & Experimental Engineering and Sciences (ICCES), 12(4), 137-143.

[19] Chang, C-L., Makar, M., Sam S.T. and Girod, B. 2010. Direction-Adaptive Partitioned Block Transform for Color Image Coding. IEEE Transactions on Image Processing, 19(7), 1740-1755.

[20] George, L. E. and Sultan, B. 2011. Image Compression Based on Wavelet, Polynomial and Quadtree. Journal of Applied Computer Science & Mathematics, 11(5), 15-20

[21] Ghadah, Al-K. and George, L. E..2013.Fast Lossless Compression of Medical Images based on Polynomial. International Journal of Computer Applications, 70(15),28-32.

[22] Maragos, P. A., Schafer, R. W. and Mersereau, R. M. 1984. Two-Dimensional Linear Predictive and Its Application to Adaptive Coding of Images. Proceedings of the IEEE international conference on Acoustics, Speech and Signal Processing, 1213-1229.

[23] Musmann, H. G., Pirsch, P. and Grallert, H. 1985. Advances in Picture Coding. Proceedings of the IEEE, 73(4), 523-548.

[24] Das, M. and Loh, N. K. 1990. New Studies on Adaptive Coding of Images using Multiplicative Autoregressive Models. 10th IEEE Region Conference on Communication, 442-446.

[25] Burgett, S. and Das, M. 1993. Predictive Image Coding using Multiresolution Multiplicative Autoregressive Models. Proceedings of the IEEE, 140(2), 127-134.

[26] Balram, N. and Moura, J. M. F. 1996. Noncausal Predictive Image Codec. IEEE Transactions on Image Processing, 5(8), 1229-1242.

[27] Su, C.K., Hsin, H.C. and Lin, S.F. 2005. Wavelet Tree Classification and Hybrid Coding for Image Compression. IEE Proceedings on Vision, Image and Signal Processing, 152(6), 752–756

[28] Iano, Y., Silva, da. And Cruz, F.S. 2006. A Fast and Efficient Hybrid FractalWavelet Image Coder. IEEE Transactions on Image Processing, 15(1), 98–105.

[29] Xu, J., Wu, F. and Zhang, W. 2009. Intra-Predictive Transforms for Block-Based Image Coding. IEEE Transactions on Signal Processing, 57(8), 3030- 3040.

[30] Gray, R. M. 2010. A Survey of Linear Predictive Coding: Part I of Linear Predictive Coding and the Internet Protocol. Foundations and Trends in Signal Processing, 3(3), 153-202.

[31] Rehna, V. J. and Kumar, M. K. J. 2011. Hybrid Approaches to Image Coding: A Review. International Journal of Advanced Computer Science and Applications (IJACSA), 2(7), 108-115.

[32] Groach, M. and Garg, A. 2012. Image Compression Algorithm. International Journal of Engineering Research and Applications (IJERA), 2(2), 560-567.