Global Journal of Computer Science and Technology
Graphics & Vision
Volume 12 Issue 13 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Novel Color Image Compression Algorithm Based on Quad tree

By A. A. El-Harby & G. M. Behery
Damietta University, Faculty of Science, New Damietta, Egypt

Abstract - This paper presents a novel algorithm having two image processing systems that have the ability to compress the colour image. The proposed systems divides the colour image into RGB components, each component is selected to be divided. The division processes of the component into blocks are based on quad tree method. For each selection, the other two components are divided using the same blocks coordinates of the selected divided component. In the first system, every block has three minimum values and three difference values. While the other system, every block has three minimum values and one average difference. From experiments, it is found that the division according to the G component is the best giving good visual quality of the compressed images with appropriate compression ratios. It is also noticed, the performance of the second system is better than the first one. The obtained compression ratios ofthe second system are between 1.3379 and 5.0495 at threshold value 0.1, and between 2.3476 and 8.9713 at threshold value 0.2.

GJCST-F Classification : I.4.2

NOVEL COLOR IMAGE COMPRESSION ALGORITHM BASED ON QUAD TREE

Novel Color Image Compression Algorithm Based on Quad tree

A. A. El-Harby [α] & G. M. Behery [α]

Abstract - This paper presents a novel algorithm having two image processing systems that have the ability to compress the colour image. The proposed systems divides the colour image into RGB components, each component is selected to be divided. The division processes of the component into blocks are based on quad tree method. For each selection, the other two components are divided using the same blocks coordinates of the selected divided component. In the first system, every block has three minimum values and three difference values. While the other system, every block has three minimum values and one average difference. From experiments, it is found that the division according to the G component is the best giving good visual quality of the compressed images with appropriate compression ratios. It is also noticed, the performance of the second system is better than the first one. The obtained compression ratios ofthe second system are between 1.3379 and 5.0495 at threshold value 0.1, and between 2.3476 and 8.9713 at threshold value 0.2.

I. Introduction

Many modern imaging systems are still producing gray-scale images, color images are more preferred due to the larger amount of information contained by them [1]. There are many compression systems were used to compress the color images; these systems include those that use mathematical transforms such as Discrete Cosine Transform (DCT) transform [2-4], neural networks [5-8], wavelet transform [9-11], fractal [12-14], quad tree systems [15-17], and others [18-22]. Data compression provides two advantages: reducing storage space and transmission time by finding the humanly imperceptible differences [23-24].

The quad tree algorithms are based on simple averages and comparisons. Quad-tree image compression is a method for splitting an image into homogenous sub-blocks. Defining the whole image as a single block, the method is performed according to some problem specific homogeneity criteria. Each block is examined to check whether it is homogenous or not. If it is not, then it will be split into four same-sized blocks. The method terminates when there is no other blocks to be split or when all blocks to be split are smaller than a pre-selected size. The minimum size of the blocks is set, to avoid over segmentation [25-28].A major advantage of the quad tree system for data compression is the simplicity of its approach. Unlike many other compression systems, a quad tree algorithm can compress images relatively quickly on a personal computer [29, 30].

Usually, distortion in images is measured by the PSNR (Peak-to-peak Signal to Noise Ratio). This ratio is often used as a quality measurement between the original and a compressed image. There is no standard way of defining distortion and PSNR for color images. The simplest way is to just average the distortions of the three RGB color components [31, 32].

In this paper, an algorithm is applied on color images. The remainder of this paper is organized as follows: in Section 2, the proposed algorithm is illustrated. Experimental results and discussion are presented in Section 3 and finally, some conclusions are addressed in Section 4.

II. Proposed Algorithm

This algorithm contains two systems based on quad tree. Each one contains three different cases. The RGB color images are represented by three components. In gray-scale image there is a high correlation between neighbor pixels. In color image, in addition to this, there is also a high correlation between color components [2,3]. Therefore, the proposed systems are applied on the all components altogether. In the first system, one component is chosen to be divided using quad tree at specified threshold value. During the dividing of this component, even if the condition of quad tree division is not verified for the other two components, they are divided simultaneously using the same coordinates and block size of the chosen component. The condition is represented by difference value is greater than threshold value. There are three cases of this system are described as follows:

1. The image is divided according to the component R using quad tree. At the same time, the dividing process is applied on the other two components G and B respectively. After the dividing is completed, the three components will have the same numbers, sizes, and coordinates of all blocks.
2. This case is similar to the first one, except, the image is divided according to the second component G, and the dividing process is applied on the other two components R and B respectively.
3. The third case is similar to the previous two cases, except, the image is divided according to the third

Author α : Damietta University, Faculty of Science, New Damietta, Egypt. E-mail : elharby@yahoo.co.uk

Global Journal of Computer Science and Technology (F) Volume XII Issue XIII Version I Year 2012

component, and the dividing process is applied on the other two components R and G respectively.

In each case, the image is divided giving the following information for the three components: number of blocks, sizes, minimum value and difference between maximum and minimum values for each block. The three components have the same coordinates and sizes for all blocks. In the three components, any block has three min values and three diff values, one for each component.

The second system is similar to the first one including the above three cases, except, the three difference values are averaged for each block. The obtained information of every block is one coordinates, one size, three minimum values and one average difference value. The two systems are illustrated in details in the next sub section.

Several quality measures can be found in the open literature of the field. The most used measures are (distortion evaluation): The mean squared errors (MSE) and the popular peak signal to noise ratio (PSNR) [2].With gray level images, the PSNR is expressed by:

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{MSE} \quad (1)$$

While, for color RGB images case [32], we have used the relation given in

$$\mathrm{PSNR} = 10 \times \log_{10} \left(\frac{255^2 \times 3}{MSE(R) + MSE(G) + MSE(B)} \right) \quad (2)$$

$$MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left(\mathrm{X}_{\mathrm{ij}} - \mathrm{y}_{\mathrm{ij}} \right)^2 \quad (3)$$

and, : are, respectively, the original and reconstructed intensities belonging to R, G or B component. The compressed image is evaluated with the compression ratio (CR) or withthebite-rateperpixel(bpp) defined as follows:

$$\mathrm{CR} = \frac{\text{Original RGB color image size in bits}}{\text{Compressed image in bits}} \quad (4)$$

$$\mathrm{bpp} = \frac{24 \text{ bits}}{\mathrm{CR}} \quad (5)$$

a) Example

This example is proposed to describe the processes of the two systems. The example is applied on a sample of color image for size 8x8x3. The R component is firstly chosen to be divided using quad tree. The other two components will be divided using the same coordinates and block size of the chosen divided component, even if the condition of quad tree division is not verified for one or both components. In the first system, each block has three min values and three difference values, one for each component. For instance, the block of coordinates (4, 0) has the three minimum values (80, 25, and 11) and the three difference values (5, 77, and 6), see Fig. 1 and table 1 for more details. While, the second system has three minimum values and one average difference. The above mentioned block of coordinates (4, 0) has the same three minimum values and one average difference value (29), see Fig. 2 and table 1 for other details.

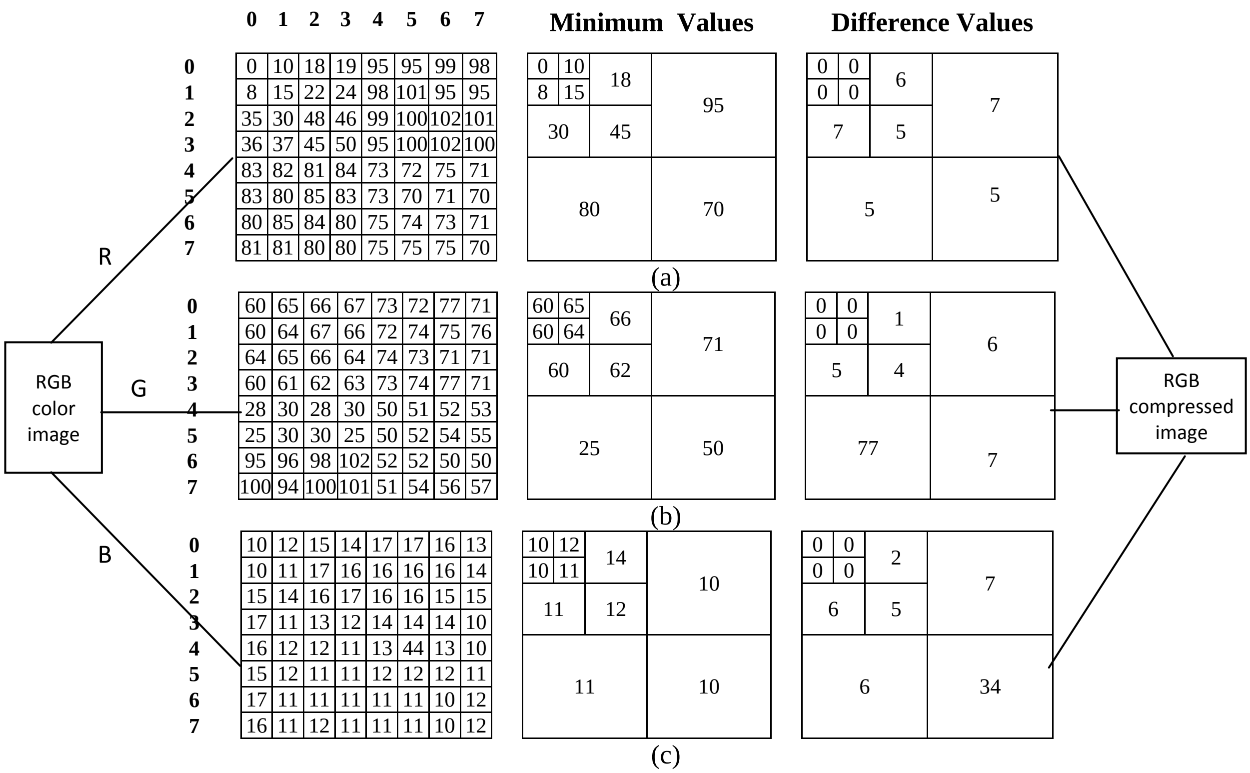

Fig. 1 : Diagram shows the processes of the first system

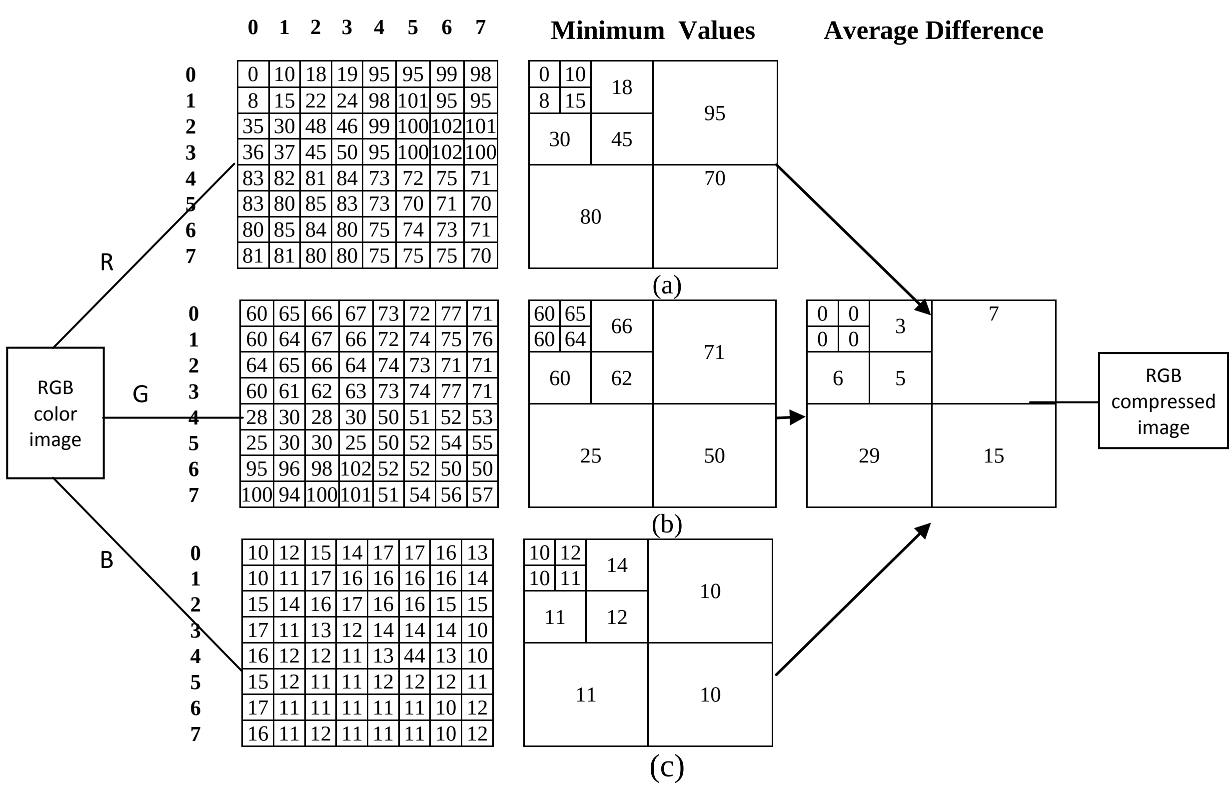

Fig. 2 : Diagram shows the processes of the second system

Global Journal of Computer Science and Technology (F) Volume XII Issue XIII Version I Year 2012

Table 1 : The obtained results of the two systems

Top-left Coord.	Size	Minimum values of both systems			Difference values of first system			Average difference of second system
		R	G	B	$S1_R$	$S1_G$	$S1_B$	
(0,0)	1	0	60	10	0	0	0	0
(0,1)	1	10	65	12	0	0	0	0
(1,0)	1	8	60	10	0	0	0	0
(1,1)	1	15	64	11	0	0	0	0
(0,2)	2	18	66	14	6	1	3	3
(2,0)	2	30	60	11	7	5	6	6
(2,2)	2	45	62	12	5	4	5	5
(0,4)	4	95	71	10	7	6	7	7
(4,0)	4	80	25	11	5	77	6	29
(4,4)	4	70	50	10	5	7	34	15

III. Experimental Results and Discussion

In order to test the performance of the two proposed systems, they are applied using the same settings on four famous color images. These images are called Splash, Lena, Sailboat, and Pepper; see Fig. (3).The dimension and the size of each image are 512x512x3 and 786432 bytes respectively. Ninety six experiments are done using the two systems, twenty four on each image. The first forty eight experiments are carried out using the first system; the threshold values are 0.1, 0.2, 0.3 and 0.5. The obtained results are shown in the first three columns of Figures (4-7). The columns S1R, S1G, and S1B represent the three components of the color image for the first system.

The other forty eight experiments are carried out using the second system using the same threshold values that are proposed with the first system. The obtained results are shown in the second three columns of the Figures (4-7). The columns S2R, S2G, and S2B are represented for the second system. The compression ratio is obtained by dividing the size of the original image file by the size of the compressed output file. From the above experiments of the two systems, Tables (2-5) show the obtained compression ratios, bpp, PSNR and number of blocks in the compressed images. All programs are written using the Matlab software.

From figures (4-9) and Tables (2-5), it can be seen the following:

- In the two systems, the number of blocks decreases when the original image has low details (for instance Splash image); see Table 5 and Figure (9).
- In the first system, the compression ratio is ranged between 1.0406:1 and 79.7275:1, while with the second system, the compression ratio is between 1.3379:1 and 102.5068:1. It is seen that the second system has the highest compression ratio.
- In the first system, the bpp is rangedbetween 23.0633 and 0.3010, while with the second system, the bpp is between 17.9381 and 0.2341. It is seen that the second system has the lowest bpp value.
- In the first system, the PSNR is rangedbetween 16.7784 and 10.7710, while with the second system, the PSNR is between 16.0773 and 15.5210.
- The visual quality of the compressed images and PSNR values are inversely proportional tothe compression ratio; see figures (4-7).
- In the two systems, the compression ratios increase when the original image has low details (for instance Splash image); see table 2 and Figure (8).
- The compressed images quality increase when the image is divided according to the component G.
- The compression ratios are proportional to the threshold values.

In Table 6, is presented comparative results among our proposed two systems and others, compression ratio is measured in terms of bpp and the image quality in terms of PSNR.

Fig. 3 : Original images

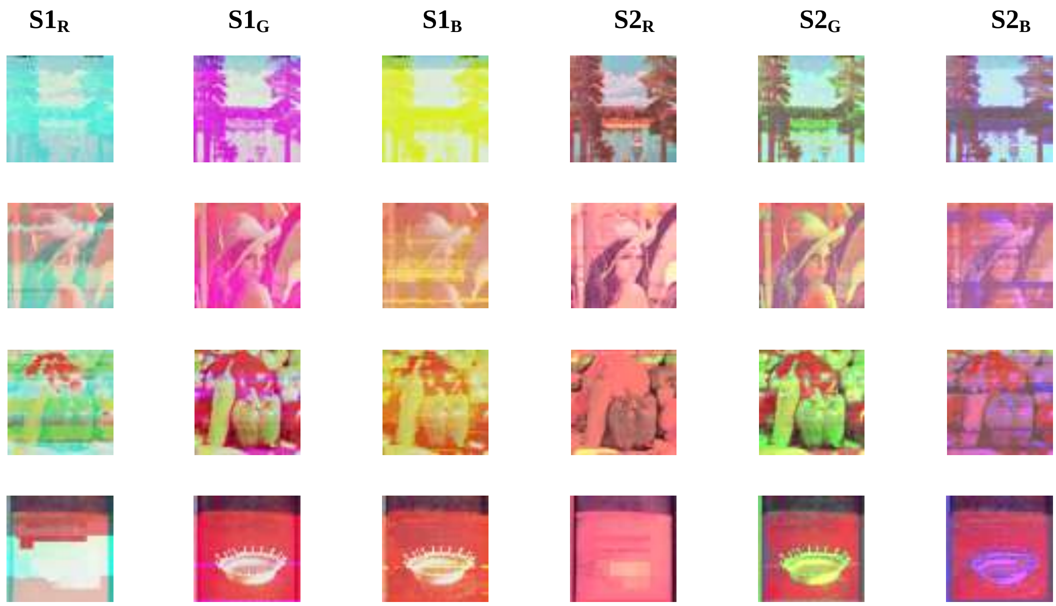

Fig. 4 : The compressed images at threshold = 0.1

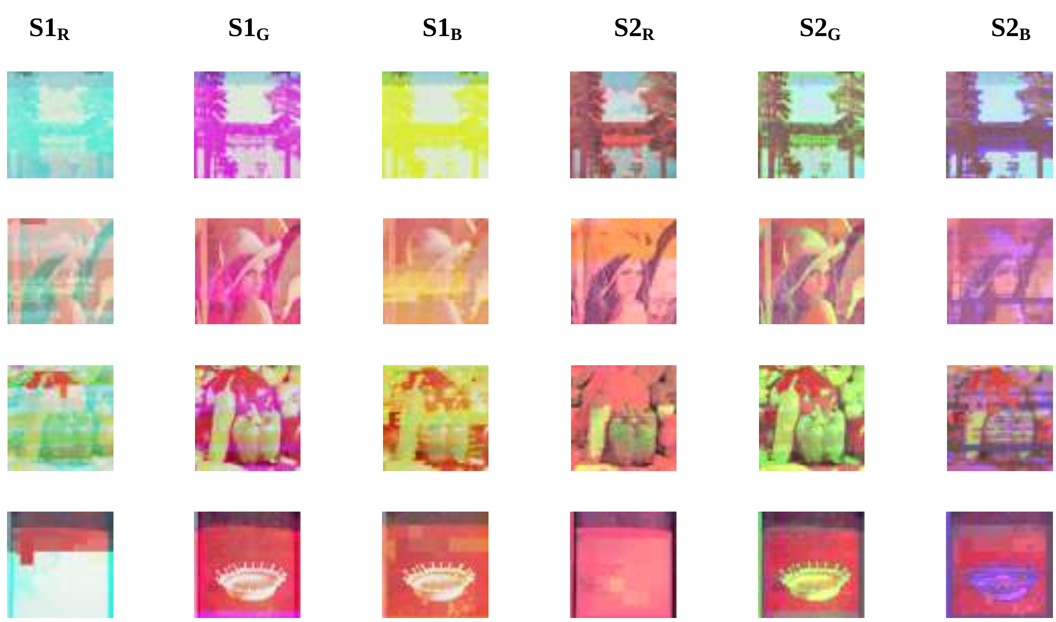

Fig. 5 : The compressed images at threshold = 0.2

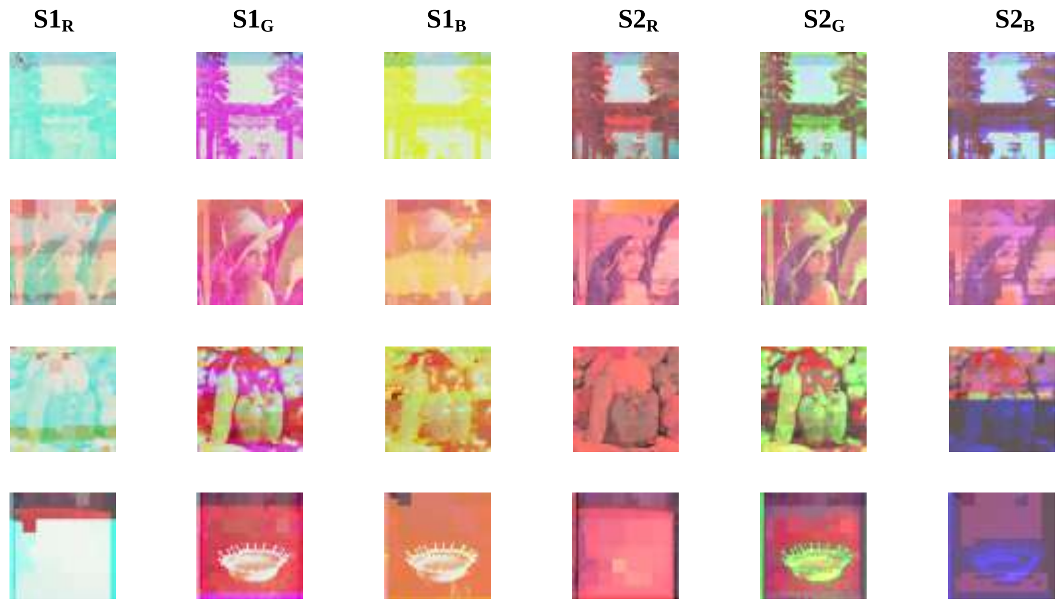

Fig. 6 : The compressed images at threshold = 0.3

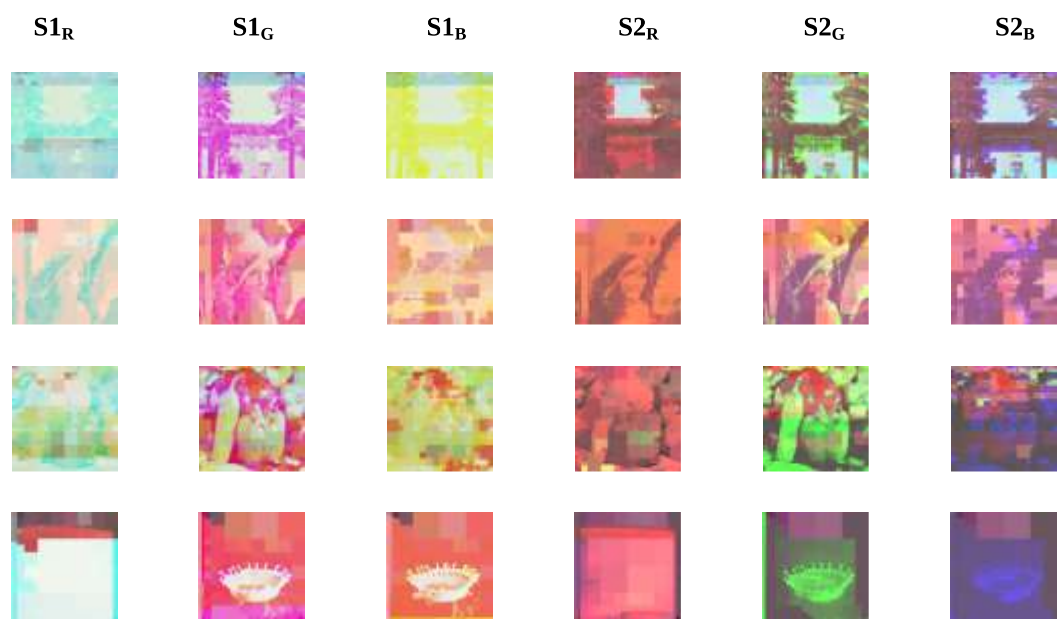

Fig. 7 : The compressed images at threshold = 0.5

Table 2 : The Compression ratios of the two systems

Image Name	Threshold = 0.1						Threshold = 0.2					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	1.5116	1.0406	1.2258	1.9435	1.3379	1.5760	3.5637	1.8259	1.9948	4.5819	2.3476	2.5648
Lena	2.9091	2.0848	2.3798	3.7403	2.6804	3.0597	5.9953	4.1028	5.1169	7.7082	5.2750	6.5789
Peppers	2.4177	2.1072	2.3802	3.1085	2.7093	3.0602	5.4146	3.7929	4.7747	6.9617	4.8766	6.1389
Splash	7.9207	3.9274	5.1788	10.1838	5.0495	6.6584	12.3142	6.9777	7.8694	15.8325	8.9713	10.1177

Image Name	Threshold = 0.3						Threshold = 0.5					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	9.8892	3.3950	3.5593	12.7147	4.3650	4.5763	36.5765	6.9000	7.4260	47.0270	8.8714	9.5477
Lena	15.1915	8.5660	12.1599	19.5319	11.0134	15.6342	39.1318	20.8299	46.5785	50.3123	26.7813	59.8867
Peppers	11.5844	6.3137	9.6789	14.8943	8.1176	12.4443	26.4471	10.8226	24.5109	34.0035	13.9147	31.5140
Splash	23.0740	10.7427	11.5021	29.6666	13.8121	14.7884	79.7275	17.5429	23.2583	102.5068	22.5552	29.9035

Table 3 : The bite-rate per pixel of the two systems

Image Name	Threshold = 0.1						Threshold = 0.2					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	15.8774	23.0633	19.5796	12.3491	17.9381	15.2285	6.7346	13.1443	12.0311	5.2380	10.2234	9.3575
Lena	8.2499	11.5120	10.0849	6.4166	8.9538	7.8438	4.0031	5.8497	4.6903	3.1136	4.5497	3.6480
Peppers	9.9267	11.3893	10.0833	7.7208	8.8583	7.8425	4.4324	6.3276	5.0265	3.4474	4.9214	3.9095
Splash	3.0300	6.1109	4.6343	2.3567	4.7529	3.6045	1.9490	3.4395	3.0498	1.5159	2.6752	2.3721

Image Name	Threshold = 0.3						Threshold = 0.5					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	2.4269	7.0692	6.7429	1.8876	5.4982	5.2444	0.6562	3.4783	3.2319	0.5103	2.7053	2.5137
Lena	1.5798	2.8018	1.9737	1.2288	2.1792	1.5351	0.6133	1.1522	0.5153	0.4770	0.8961	0.4008
Peppers	2.0717	3.8013	2.4796	1.6114	2.9565	1.9286	0.9075	2.2176	0.9792	0.7058	1.7248	0.7616
Splash	1.0401	2.2341	2.0866	0.8090	1.7376	1.6229	0.3010	1.3681	1.0319	0.2341	1.0641	0.8026

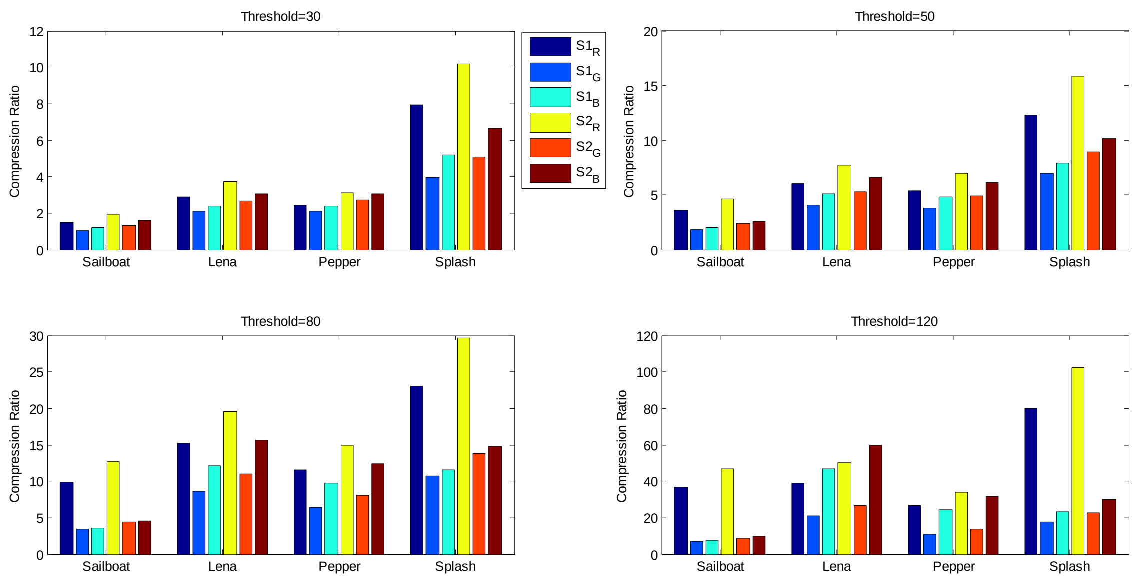

Fig. 8 : The compression ratios of the two systems at the proposed threshold values

Table 4 : PSNR for two systems

Image Name	Threshold = 0.1						Threshold = 0.2					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	15.4361	16.7784	16.7262	18.8796	16.0773	15.9801	13.7661	15.0063	15.0697	17.4199	16.3393	16.1908
Lena	18.7293	19.6201	18.9964	16.4421	17.5186	16.9536	16.8739	19.3587	17.0483	16.8418	18.2939	16.8419
Peppers	16.7775	20.0167	18.9491	15.2504	17.6726	14.8351	15.0769	17.5252	17.2620	15.6042	16.9327	15.4747
Splash	16.7762	22.3761	21.5052	15.5125	17.8204	17.0871	14.1374	19.5457	18.8715	15.4450	18.1920	15.7507

Image Name	Threshold = 0.3						Threshold = 0.5					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	11.8527	13.5657	13.3850	16.5140	16.6293	16.3797	10.5626	12.1440	11.7933	15.0449	17.0646	16.6874
Lena	14.8542	17.0269	14.6512	16.8986	17.7427	16.9181	12.1920	14.7676	12.8821	18.3826	17.3689	17.0477
Peppers	12.2413	15.9231	15.3282	15.2964	16.9355	14.6655	11.5015	14.2065	13.5195	15.8796	16.2712	14.6080
Splash	11.6013	17.3963	15.7024	15.4722	16.7581	13.3793	10.7710	13.7554	14.7283	15.5210	13.8537	13.0139

Table 5 : The number of blocks for the two systems

Image Name	Threshold = 0.1						Threshold = 0.2					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	30528	59800	50796	30528	59800	50796	7548	27240	27416	7548	27240	27416
Lena	13972	22924	19012	13972	22924	19012	3912	8368	6536	3912	8368	6536
Peppers	15556	22676	17548	15556	22676	17548	5612	10448	6888	5612	10448	6888
Splash	4068	9248	8416	4068	9248	8416	2592	5940	5720	2592	5940	5720

Image Name	Threshold = 0.3						Threshold = 0.5					
	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$	$S1_R$	$S1_G$	$S1_B$	$S2_R$	$S2_G$	$S2_B$
Sailboat	1084	9212	11152	1084	9212	11152	148	1756	1784	148	1756	1784
Lena	620	2324	1536	620	2324	1536	32	432	64	32	432	64
Peppers	2032	4964	2348	2032	4964	2348	732	2216	588	732	2216	588
Splash	1252	3868	3496	1252	3868	3496	380	1820	1136	380	1820	1136

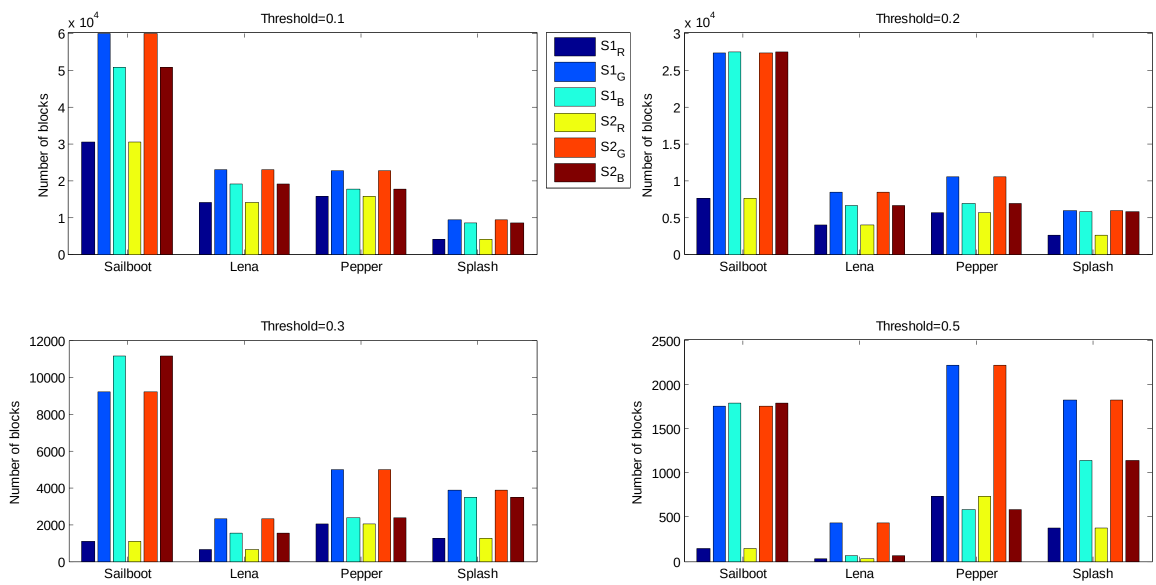

Fig. 9 : The number of blocks for the two systems at the proposed threshold values

Table 6 : Comparison between the proposed systems and others

Image	Proposed				Other Systems							
	System1		System 2		YCbCr [2]		CBTC-PF[32]		JPEG [32]		NNET [33]	
	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR	Bpp	PSNR	bpp
Lena	17.0483	0.5153	16.8419	0.4008	31.97	0.8101	31.93	1.17	32.7729	1.0073	21.9876	5.8
Pepper	11.5015	0.9075	15.8796	0.7058	30.059	0.809	30.15	1.5	30.47	1.47	---	---

IV. Conclusion

This paper presents two efficient systems that have the ability to compress colour images in easy way. The division processes of image into blocks for the two systems are based on quad tree. At the dividing of one component, the other two components are divided using the same division even if the condition of quad tree division is not verified for them. After the division process is completed, the three components will have the same number and size of blocks. During the experimental results, the compression ratios, bit rate per pixel and PSNR are computed. The compression ratios of images are increased by increasing the value of threshold while the quality of the compressed images may be decreased. It was also noticed, the division according to the G component is the best giving good quality of the compressed images with appropriate compression ratios, and the performance of the second system is better than the first one. The compression ratios of the second system are ranged between 0.25 and 0.80 at threshold value 0.1, and between 0.78 and 0.94 at threshold value 0.2.

References Références Referencias

1. Abadpour a, S. Kasae, "New PCA-based Compression Method for Natural Color Images", IPM Workshop on Computer Vision, 7 May 2005.
2. Fouzi Douak, Redha Benzid, Nabil Benoudjit, "Color image compression algorithm based on the DCT transform combined to an adaptive block scanning", AEU - International Journal of Electronics and Communications, Vol. 65(1), 2011, pp. 16-26.
3. Amit Phadikar, Santi P. Maity, "Quality access control of compressed color images using data hiding", AEU - International Journal of Electronics and Communications, Vol. 64(9), 2010, pp. 833-843.
4. Evgeny Gershikov, Emilia Lavi-Burlak, Moshe Porat, Correlation-based approach to color image compression Signal Processing: Image Communication, Volume 22, Issue 9, October 2007, pp. 719-733.
5. B. Sowmya, B. Sheela Rani, Colour image segmentation using fuzzy clustering systems and competitive neural network, Applied Soft Computing, Vol. 11(3), 2011, pp. 3170-3178.
6. S. Immanuel Alex Pandian, and J. Anitha, "A Neural Network Approach for Color Image Compression in Transform Domain", International Journal of Recent Trends in Engineering, Vol. 2(2), 2009, pp. 152-154.
7. W. Kurdthongmee, "A novel hardware-oriented Kohonen SOM image compression algorithm and its FPGA implementation", Journal of Systems Architecture, Vol. 54(10), 2008, pp. 983-994.
8. N. Sudha, "An ASIC implementation of Kohonen's map based colour image compression", Real-Time Imaging, Vol. 10(1), 2004, pp. 31-39.
9. Joewono Widjaja, "Noisy face recognition using compression-based joint wavelet-transform correlator ", Optics Communications, Vol. 285(6), 15 2012, pp. 1029-1034.
10. Hong Pan, Li-Zuo Jin, Xiao-Hui Yuan, Si-Yu Xia, Liang-Zheng Xia, "Context-based embedded image compression using binary wavelet transform", Image and Vision Computing, Vol. 28(6), 2010, pp. 991-1002.
11. M S Joshi, R R Manthalhkar, Y V Joshi, "Color Image Compression Using Wavelet and Ridgelet Transform", Proceedings of 3rd IEEE International Conference on Image Processing (2010), Vol. 1, pp. 541-544.
12. George, L.E.; Al-Hilo, E.A.(2011) "Speeding-up Fractal Color Image Compression Using Moments Features Based on Symmetry Predictor", Information Technology: New Generations (ITNG), 2011 Eighth International Conference on Digital Object Identifier, pp. 508-513.
13. George, L.E.; Al-Hilo, E.A. (2010) "Isometric process behavior in fractal color image compression by zero-mean method", Computer Engineering and Technology (ICCET), 2010 2nd International Conference on Vol. 6, pp. 692-697.
14. Ghosh, S.K.; Mukherjee, J. (2004) "A novel fractal technique for color image compression", India Annual Conference, Proceedings of the IEEE INDICON 2004, pp. 170-173.
15. Helmut Jürgensen, Ludwig Staiger, Hideki Yamasaki (2007) "Finite automata encoding geometric figures", Theoretical Computer Science, Vol. 381(1-3), pp. 33-43.
16. Yung-Kuan Chan (2004) "Block image retrieval based on a compressed linear quadtree", Image and Vision Computing, Vol. 22(5), pp. 391-397.
17. N. Udaya Kumar1 and K. Padma Vasavi (2011) " AN EFFICIENT COLOR CODING SCHEME FOR COLOR IMAGES USING QUAD TREE DECOMPOSITION AND COLOR PALETTES", World Journal of Science and Technology, Vol. 1(8), pp.26-31.
18. Yuancheng Li, Qiu Yang, Runhai Jiao, "Image compression scheme based on curvelet transform and support vector machine", Expert Systems with Applications, Vol. 37(4), 2010, pp. 3063-3069.
19. Bibhas Chandra Dhara, Bhabatosh Chanda, "Color image compression based on block truncation coding using pattern fitting principle", Pattern Recognition, Vol. 40(9), 2007, pp. 2408-2417.
20. G. Sreelekha, P.S. Sathidevi, "An HVS based adaptive quantization scheme for the compression of color images", Digital Signal Processing, Vol. 20(4), 2010, pp. 1129-1149.
21. Satish Kumar Singh, Shishir Kumar (2011) "Novel adaptive color space transform and application to image compression", Signal Processing: Image Communication, Vol. 26(10), pp. 662-672.
22. Pan-Pan Niu, Xiang-Yang Wang, Yi-Ping Yang, Ming-Yu Lu (2011) "A novel color image watermarking scheme in nonsampled contourlet-domain", Expert Systems with Applications, Vol. 38(3), pp. 2081-2098.
23. Kuo-Liang Chung, Hsu-Lien Huang, Hsueh-I Lu, (2004) "Efficient region segmentation on compressed gray images using quadtree and shading representation", Pattern Recognition, Vol. 37(8), pp. 1591-1605.
24. Guangtao Zhai, Weisi Lin, Jianfei Cai, Xiaokang Yang, Wenjun Zhang (2009) "Efficient quadtree based block-shift filtering for deblocking and deranging", Journal of Visual Communication and Image Representation, Vol. 20(8), pp. 595-607.
25. Kazuya Sasazaki, Sato Saga, Junji Maeda, Yukinori Suzuki (2008) "Vector quantization of images with variable block size", Applied Soft Computing, Vol. 8(1), pp. 634-645.
26. Kutil, R, and Gfrerer, C. (2011) "A generalization of quad-trees applied to shape coding", Systems,

Signals and Image Processing (IWSSIP), 2011 18th International Conference on, pp. 1-4.

27. M Anand Kumar1, D Demudubabu 2, Ch Mohan Babu (2012) "Image compression using area sub-division algorithm relying on Quadtree ", Asian Journal of Computer Science and Information Technology, Vol. 2(3), pp. 19 –22.
28. El-Harby A.A., and Behery G.M., (2008) "Qualitative Image Compression Algorithm Relying on Quadtree", International Journal on Graphics, Vision and Image processing (GVIP), Vol. 8(3), pp. 41-50.
29. Evgeny Gershikov, Moshe Porat (2007) "On color transforms and bit allocation for optimal subband image compression", Signal Processing: Image Communication, Vol. 22, pp. 1-18.
30. Bibhas Chandra Dhara, and Bhabatosh Chand (2007) "Color image compression based on block truncation coding using pattern fitting principle", Pattern Recognition, Vol. 40, pp. 2408 -2417.
31. Ashikur Rahman A. K. M. and Chowdhury Mofizur Rahman (2003) "A New Approach for Compressing Color Images using Neural Network", CIMCA Proceedings 12-14 February 2003, Vienna – Austria, pp. 315-326.