International Journal of Academic Engineering Research (IJAER)
ISSN: 2000-001X
Vol. 2 Issue 10, October – 2018, Pages: 19-27

Bitplanes Block Based Lossy Image Compression

Abdelatief H. Abouali
HICI, Computer Dept., El-Shorouk Academy, Cairo, Egypt,
Email: dr.abdullatif.hussein@sha.edu.eg

Abstract: *In a former paper [21], an exact image compression based on bit-planes blocking was proposed. The proposed algorithm uses two bit codes for block representation. The outcome of the encoding process is two streams: Main Bit Stream, MBS and Residual Bit Stream, RBS. The algorithm core is searching for the greatest block of Unicode to encode in main stream and if not found until size of two by two then it will be kept as is in residual stream. In this paper, a lossy version of that algorithm is presented. The change in the base algorithm is in the definition of the unary-code-block is eased to be above certain percent. The percent is varied from plane to another as their contribution to image power varies. The testing of the proposed algorithm shows comparable results. Image degradations seems restorable even for high compression ratios.*

Keywords: bit-planes, image blocks, exact-image compression, encoding, decoding.

1. Introduction

The success of multimedia and image processing based systems, in many cases, is highly tightened to effective encoding to digital images. Demands and the volume of digital images used in systems currently in use within the domains of : education, security, social medial, health care, retail storage, industry quality assurance, entertainment, law enforcement and many others is huge and subject to grow [1-3]. Therefore, effective storage, processing, transmitting, and recall needed for the development process to continue. To date, human effective storage, processing, recognition, indexing, and recall is far above all developed methodologies and devices man made.

The encoding process is an effective representation, in computer vision systems, that increases system capacity to store, access, exchange, and process digital images. Image encoding is achieved by the removal of one or more of the basic image data redundancies: Coding, Interpixel, and Psychovisual [4-5]. Coding redundancy is due to the use of non-optimal code words. Interpixel redundancy results from correlations between image pixels. Psychovisual redundancy is the existence of data that is insignificant to the human visual system (i.e. visually non-essential). Encoding techniques require decoding process to retrieve the compressed image for further use by applications. In video compression, association of frame images, abstraction, and relationships adds more significant encoding step to sets of frame images [6].

Image compression techniques are exact and Lossy[7]. The exact compression techniques assure the retrieval of the decompressed image typical as the original. Lossy compression techniques allow controlled loss of power. The exact image compression techniques include, pixel packing, run-length, Huffman, LZW, arithmetic and Area coding. Lossy techniques includes Transformation Coding, differential, Vector Quantization, object based, Fractal, Block truncation coding, and Sub band coding [8-12]. Good encoding scheme means: low order of algorithm complexity for both encoder and decoder, high signal to noise ratio, high compression ratio, ability to decode at varieties of scales without additional distortion, parallelization ability, as well as the ease to implement software and/or hardware. The well-known encoding algorithms, such as JPEG and MPG, employs a set of basic encoding schemes such as Huffman, differential, quantization, and run-length [13-14].

Block based image compression schemes are numerous as dealing with whole image as a processing unit costly. Blocking coincide with images reality that are connected blocks/labels. From the well-known block based algorithms JPEG, and fractal. In JPEG the blocks are transformed to frequency domain, using DCT, followed by reduction to insignificant blocks, then quantization followed by differential and entropy encoding. Blocking offers JPEG two main advantages low cost transformation and reduces the possibilities of having higher frequencies components. Fractal image encoding based on establishing similarities between small block, ranges, and larger blocks, domains. The blocks similarities enabled the use of the iterative function systems which is the base of fractal encoding. Other blocking schemes could be found in [14-17].

Image Bit-plane is a bit pixel decomposition of an image matrix. Therefor a gray image of n-bit gray resolution contains n-planes. Least Significant Bit Planes, LSBP, contain less significant information compared Most Significant Bit Planes, MSBP. Figure (1) shows the original versus the bit-planes for 8-bit boy and Lena pictures. The bit-planes was a rich topic for both enhancement and compression algorithms as plane-pixels contains only two values 0, and 1. So, The Run Length Encoding, arithmetic and progressive transition are used in bit-planes taking the advantages of binary and similarities of adjacent bits within the same plane [18-20].

International Journal of Academic Engineering Research (IJAER)
ISSN: 2000-001X
Vol. 2 Issue 10, October – 2018, Pages: 19-27

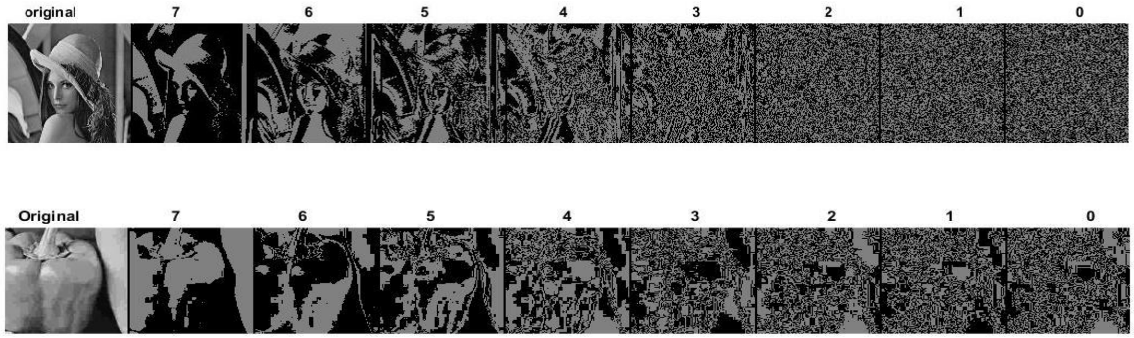

Figure (1) Lena, pepper original and bit-planes for 8-bit gray

In this paper, a lossy version of former proposed exact compression/decompression algorithm based on successive block test then encode or divide process is proposed. The algorithm uses two-bit structural encoding together with residual block storage. The encoding is two phases. In the first phase three codes are used and in the second phase third code is consumed in an adaptive run length encoding, for more optimization.

The rest of the paper is organized as follows: The proposed basic encoding/decoding is described in section 2. Tests, results and discussion are in section 3. Section 4 is the study conclusion.

2. THE PROPOSED ENCODING/DECODING ALGORITHMS

The proposed algorithm is a lossy version of a former proposed exact encoding/decoding algorithm [21]. The exact encoding and decoding algorithm are in appendix A, and B consequently. The modification is limited to the encoding process. So, the decoding process is the same for both exact and lossy. The change to the encoding is in the consideration of block all zeros and block all one. The consideration is eased from all one's and all zero's to above certain percent. Planes contributes differently to picture energy, figure (1). Consequently, percent's considered to vary from a plane to another. That is, for a given square block of size, SB with non-null count (BNNC) and block one's count (B1C) the block is considers all zeros only if: -

$$BNNC > 0, \text{ and}$$

$$\frac{(BNNC - B1C)}{BNNC} \geq p_i \qquad (1)$$

Also, a block is considered all one's only if: -

$$BNNC > 0, \text{ and}$$

$$\frac{(B1C)}{BNNC} \geq p_i \qquad (2)$$

p_i is plane allowance percent. That is, in the lossy version encoding algorithm the switch case check 'all zeros' refers to equation (1). Similarly, check of all one's refers to equation (2). Table (1) shows how much of accepted different bits for different block sizes on different percent's.

Corollary(1): The allowance percent p_i of 0.5 at a plane will lead to the plane encoded as only two-bit either '00' or '11'. That comes from the fact that a plane either one code is more than the second, or equal to. In both cases it will be considered unary code block. Consequently, RBS will be null.

Corollary(2): The percent 0.5< p_i <0.75 at a plane implies that the RBS nibbles will contain only the hex codes {3,5,6,9,A,C}.

In the encoding process, blocks coded or split process until 2x2 size. At this size with that percent range one bit is allowed to have different value for block to be encoded as unary-code block. So, residual will be the case of codes equal counts.

Table (1): Accepted different codes for a block size per percent

	2x2	4x4	8x8	16x16	32x32	64x64	132x132	256x256
97%	0	0	1	5	20	81	327	1310
94%	0	0	3	12	51	204	819	3276
91%	0	1	5	20	81	327	1310	5242
88%	0	1	7	28	112	450	1802	7208
85%	0	2	8	35	143	573	2293	9175
82%	0	2	10	43	174	696	2785	11141
79%	0	3	12	51	204	819	3276	13107
76%	0	3	14	58	235	942	3768	15073
73%	1	4	16	66	266	1064	4259	17039
70%	1	4	18	74	296	1187	4751	19005
67%	1	5	20	81	327	1310	5242	20971
64%	1	5	22	89	358	1433	5734	22937
61%	1	6	24	97	389	1556	6225	24903
58%	1	6	26	104	419	1679	6717	26869
55%	1	7	28	112	450	1802	7208	28835
52%	1	7	30	120	481	1925	7700	30801

3. EXPERIMINTAL RESULTS AND DISCSSION

The used test set of images is the same as of [21], aerial, cameraman, woman-house, Barbara, Lena, and house, Figure (2). The colored ones changed to gray using 'Matlab' 'rgb2gray' function as the performance noted to be same for both colored and gray. The percent array in our experiments

International Journal of Academic Engineering Research (IJAER)
ISSN: 2000-001X
Vol. 2 Issue 10, October – 2018, Pages: 19-27

was based on the fact that loss on the higher planes implies great loss in image power. The outer high planes, 7 and 6, assigned high and equal percent's. Considering the implications of Corollaries(1,2), other plane i percent is set in accordance with the rule $p_i = \max(\frac{p_{i-1}}{2}, 0.52)$. Figure (3) shows the ease percent for planes of our experiments. The highest planes percent's varied from 0.99 to 0.80 with 0.005 decrease step.

Assuming $f(x, y)$ is original image matrix of size , N, M are the matrix dimension, $q_{\max}$ is the peak gray, and $fp(x, y)$ is decoded image matrix. Then, the used metrics in our study are: -

Compression Ratio,

$$CR = \frac{original - image - size}{encoded - image - size}$$

Mean Square Error,

$$MSE = \frac{\sum_y \sum_x (f(x, y) - fp(x, y))^2}{N * M}$$

Normalized Root Mean Square Error,

$$NRMSE = \sqrt{\frac{\sum_y \sum_x (f(x, y) - fp(x, y))^2}{\sum_y \sum_x (f(x, y))^2}}$$

Peak Signal to Noise Ratio

$$PSNR = 10 \log_{10}((q_{\max})^2 / MSE) \quad db$$

Aerial

Cameraman

Woman-House

Lena

House

Barbara

Figure (2) set of images used in the study

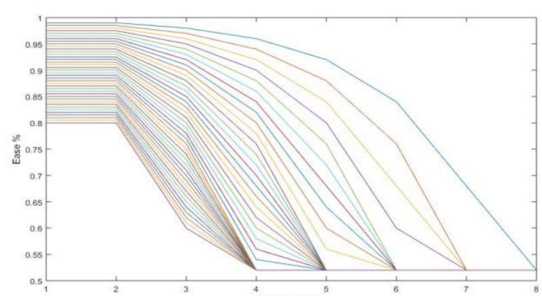

Figure (3) Experiments ease percent per planes.

The former setup made up 39 experiments done on the six test set images mentioned before. The results are summarized in figures (4) and (5). The figures are the compression ratio against NRMSE, and PSNR. The overall shows house was the best and the aerial the worst. The rest four performance is between the two. The overall performance is comparable with reported in [22-24].

International Journal of Academic Engineering Research (IJAER)
ISSN: 2000-001X
Vol. 2 Issue 10, October – 2018, Pages: 19-27

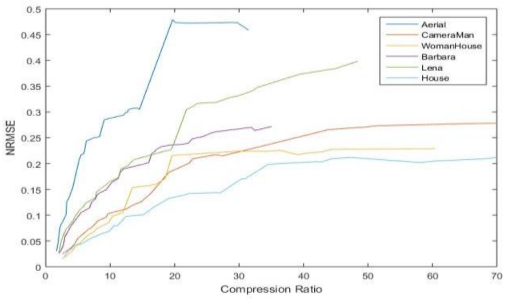

Figure (4) compression ratio against NRMSE

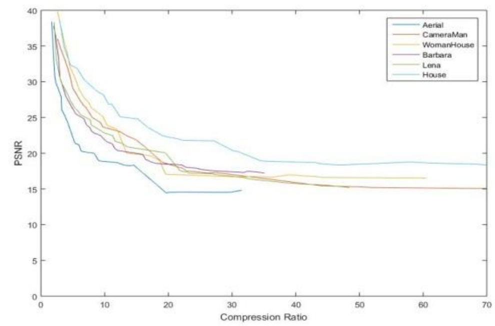

Figure (5) Compression ratio against PSNR

For more focus on the performance figure (6) presents the first seven experiments on each of the test set images. Table (2) presents numerically the corresponding compression ratios. The compression ratios compared to the visual noted distortion is also comparable to recent researches [16] [23]. There is no blocking problem or areas of severe distortion could be noted. The nature of noise looks to be within the reach of filters. Moreover, filters could be designed to partially remove the noise.

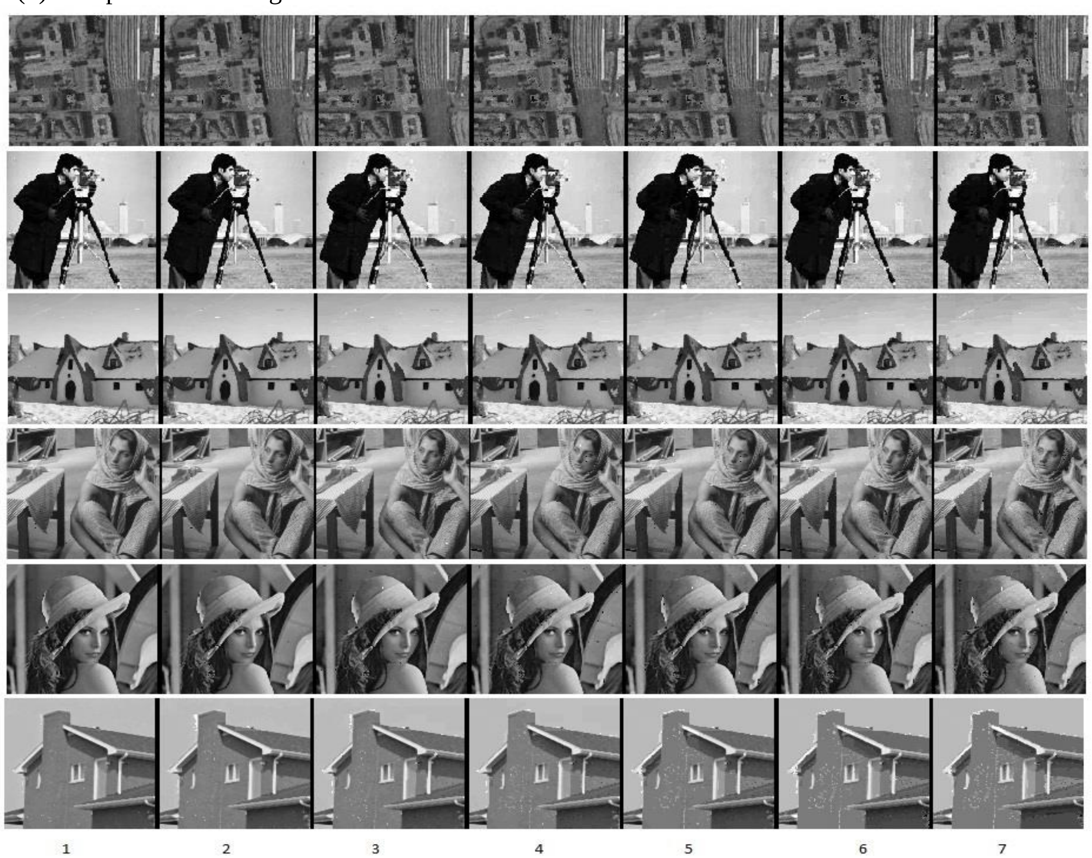

Figure (6) Decoded images for the first 7 experiments

International Journal of Academic Engineering Research (IJAER)
ISSN: 2000-001X
Vol. 2 Issue 10, October – 2018, Pages: 19-27

Table (2) Compression Ratio’s associated with figure (6)

	1	2	3	4	5	6	7
Aerial	1.672	1.892	2.171	2.347	2.573	3.009	3.154
Camera-Man	2.600	3.153	3.902	4.451	4.981	6.085	6.700
Woman’s House	2.569	3.058	3.681	4.064	4.522	5.443	5.858
Barbara	1.987	2.298	2.714	2.983	3.297	3.894	4.129
Lena	2.061	2.415	2.891	3.195	3.566	4.248	4.585
House	2.910	3.598	4.626	5.657	6.776	8.742	9.807

For more clarification to the performance, compression ratios more or equal to (3,5,10,15,20) are selected and the decoded images are in figure (7). The images degradation for even the 20+ is relatively low. There are no significant blocking effects on the recalled images. The results are comparable to the reported in [22-24].

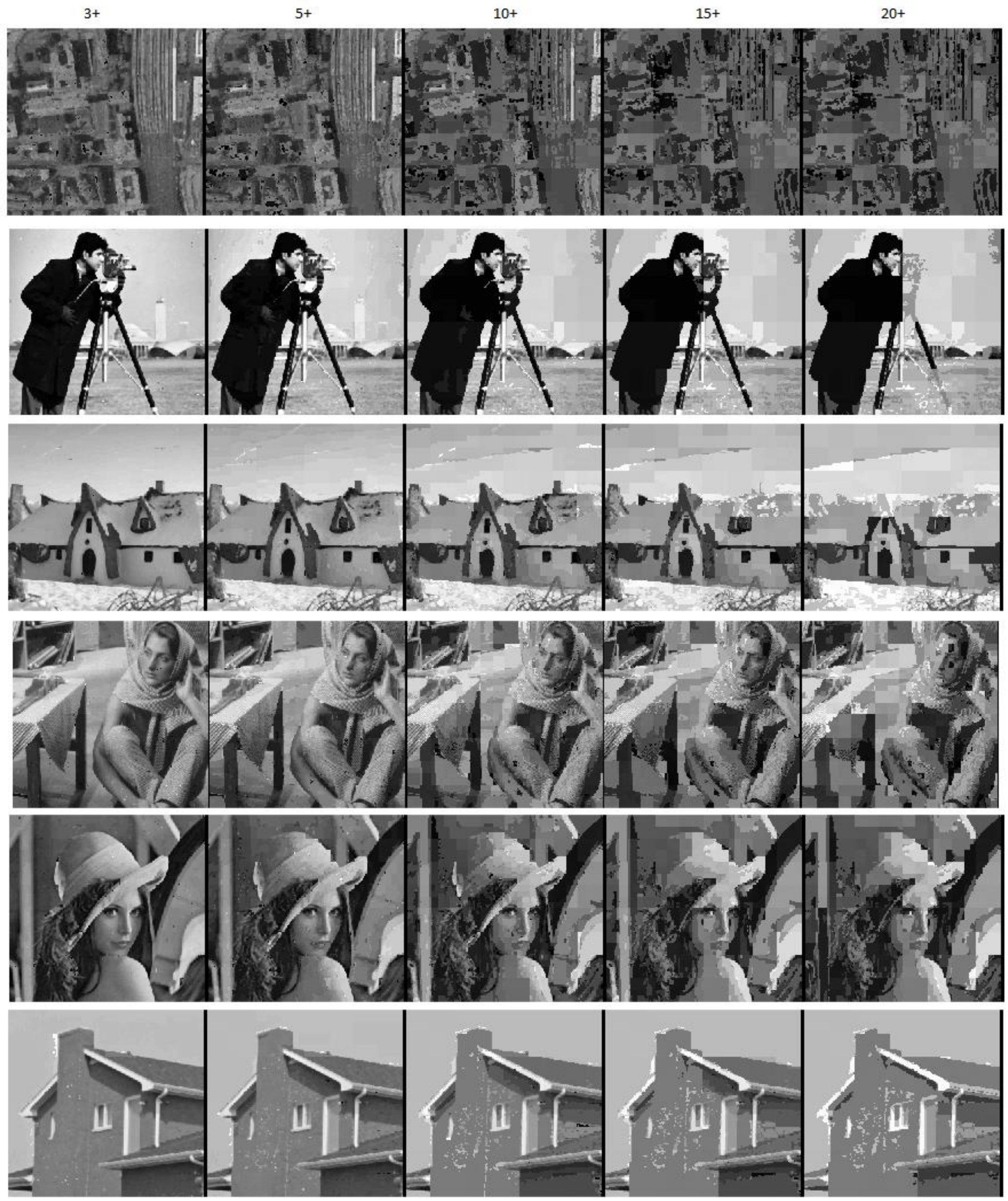

Figure (7) decoded images at compression ratios more than (3,5,10,15,20)

To sum up the results in numerical form for the 39 by 6 experiments, the two extreme perfomance cases were selected and one of the average performnce are displayed in Table (3). The table show that a significant compression ratios could be reached with comparable NRMSE. Also, there are many cases where there is a significant increase in compression ratios while insegnifcant changes in images degradation. Keeping in mind that the algorithm does not include neither transformation nor multiple iterations that makes it comparable and appealing compared to others.

Discussion

The presented algorithm offers a lossy compression technique with significant features compared to others. The algorithm performance is comparable to others. The algorithm is of minor differential to the exact version. In fact, the exact is a special case if the percent vector is set to 100%. All the image properties positively affect compression ratio discussed in [21] applies since the core algorithm is the same. Also, preprocessing with low pass filters positively adds to algorithm performance

Table 1: Compression ratio NRMSE for Worest, Best, and Average

Aerial		House		Camera Man	
CR	NRMSE	CR	NRMSE	CR	NRMSE
1.6728	0.0304	2.910609	0.020399	2.600351	0.02498
1.8922	0.0455	3.597839	0.031459	3.153668	0.029818
2.1716	0.0741	4.626655	0.042435	3.902665	0.036539
2.3477	0.0818	5.656909	0.044654	4.451493	0.043716
2.5736	0.0862	6.776373	0.053694	4.981358	0.055194
3.0090	0.0973	8.741921	0.064381	6.084980	0.066724
3.1549	0.1016	9.807658	0.068514	6.700679	0.073272
3.2454	0.1255	10.60624	0.079719	7.062354	0.075402
3.6107	0.1340	11.10826	0.079535	8.047274	0.08861
3.9924	0.1467	11.97551	0.090987	9.327474	0.09604
4.1626	0.1525	12.43567	0.097699	9.715153	0.10329
5.2331	0.2098	15.134461	0.100695	12.41359	0.11142
5.3096	0.2116	16.774532	0.115704	13.10982	0.11634
5.4645	0.2168	19.130409	0.132555	13.58012	0.11977
5.8733	0.2188	19.667191	0.134330	14.54012	0.12450
6.0075	0.2239	21.39776	0.138675	14.94081	0.12713
6.2881	0.2439	22.238	0.142236	16.3467	0.14284
7.2050	0.2483	24.9233	0.143216	19.0907	0.18284
7.3697	0.2498	26.5032	0.143577	22.27127	0.20088
7.7786	0.2503	27.073	0.14317	22.77631	0.2091
8.3605	0.2526	30.333	0.169856	26.2039	0.2168
8.8570	0.2755	30.88	0.17039	26.41914	0.21689
9.0809	0.2851	34.551	0.198642	27.42235	0.21492
12.1069	0.2941	41.7626	0.2035	43.902	0.2657
12.3688	0.2990	42.8164	0.20317	47.4468	0.26891
12.6646	0.2993	44.326	0.20919	49.7379	0.27094
12.8499	0.3049	44.326	0.209199	51.73554	0.273481
13.6313	0.3074	46.1034	0.210977	65.40518	0.277392
14.2408	0.3075	47.15244	0.211869	69.06705	0.278211
14.5333	0.3049	47.26295	0.211979	72.70669	0.27795
19.6916	0.4790	58.15084	0.201913	91.98035	0.282861
20.0761	0.4736	58.7174	0.202904	96.00585	0.284234
20.2131	0.4736	60.92121	0.205302	98.08942	0.285546
21.3681	0.4724	67.47593	0.208756	104.5856	0.286996
23.0913	0.4721	69.27695	0.210641	106.1526	0.287084
23.3942	0.4724	70.364783	0.213492	108.18984	0.289048
28.5544	0.4736	90.974839	0.2210526	121.50359	0.290525
29.7232	0.4732	90.974839	0.2210526	137.42804	0.293213
31.5115	0.4584	115.0258	0.2216756	266.00101	0.299966

More complex adaption could be considered such as dropping RBS for equal code cases. In this case, the neighbor blocks could be used heuristically to infer the codes position. Also, successive plane encoding could consider more detailed process. Such as, while encoding planes in sequence starting from the MSBP and for block B in plane i there could be bits that known it will be recovered different. The difference could be 0 recovered as 1 or the opposite. Case of 0 recovered as 1 set all corresponding bits of planes i-1 to 0 to zero and vice versa. This raises the probability of having significant distortion and will enhance the apparent patches that exist in the recovered images Figure (7). Also, considering more detailed percent vector that allows per block size per plane percent could reduce the apparent patches. These changes and others could be the considered in further studies since it requires more investigation and specifications.

4. CONCLUSION

A block based lossy image compression is proposed. The proposed algorithm is generalization to former presented algorithm. The generalization is easing the cases of blocks consideration as unary-coded. The algorithm was tested against six well known gray 256 level images. The easing percent used is plane dependent. That is, ease percent array of size equal to the image number of bit-planes. The tests used variety of easing percent to allow range of compression ratios. The performance of the algorithm found to be comparable with others.

REFERENCES

[1] Tinku Acharya and Ajoy K. Ray, Image Processing: Principles and applications, Wiley-Interscience, 2005

[2] V.P. Baligar, L.M. Patnaik, G.R. Nagabhushana, High compression and low order linear predictor for lossless coding of grayscale images, Image Vis. Comput. 21 (6),2003.

[3] D. Salomon, Data Compression: The Complete Reference, third ed., Springer, New York, 2004.

[4] D. Salomon, A Guide to Data Compression Methods, Springer, NewYork, 2002.

[5] Sonal, Dinesh Kumar, A study of various image compression techniques, Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA, June 13-15, 2007

[6] Jens-Rainer Ohm, Gary J. Sullivan, and Heiko Schwarz 'Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)', IEEE Transactions on Circuits and Systems for Video Technology (Volume: 22, Issue: 12, Dec. 2012), pages 1669 - 1684

[7] A.J. Hussain, Ali Al-Fayadh, and Naeem Radic, Image compression techniques: A survey in lossless and lossy algorithms, Neurocomputing, volume 300 26 July 2018, Pages 44-69

[8] Rafael Gonzalez, Richard Woods, Digital Image Processing 3rd edition, Pearson Prentice Hall, 2008.

[9] Tzu-Chuen Lu and Ching-Yun Chang, A Survey of VQ Codebook generation, *Journal of Information Hiding and Multimedia Signal Processing*, Volume 1, Number 3, July 2010.

[10] ABDELATIEF. H ABOUALI," OBJECT-BASED VQ FOR IMAGE COMPRESSION ", Ain Shams Engineering Journal, Vol. 6, Issue 1 (2015) pp. 211-216

[11] A. H. Abou-Ali, S. M. Abd-El-Moetty, B. Earl Wells, "RE-CONFIGURABLE HARDWARE BASED FRACTAL NEURAL PROCESSOR", The international conference on parallel processing, PDCS-2006, 2006, CA USA.

[12] Priyadarshini K S, G S Sharvani, Prapulla S B, A SURVEY ON PARALLEL COMPUTING OF IMAGE COMPRESSION ALGORITHMS JPEG and Fractal Image Compression, International Conference on Computational Systems for Health & Sustainability(CSFHS) 17-18 April 2015

[13] A. Murat Tekalp, Digital Video Processing 2nd, Prentice Hall NJ, USA 2015

[14] Mehwish Rehman, Muhammad Sharif and Mudassar Raza, Image Compression: A Survey, Research Journal of Applied Sciences, Engineering and Technology Maxwell, 2014

[15]- Surendar Chandra, and Windsor W. Hsu, Lossless Medical Image Compression in a Block-Based Storage System, march 2014 Data Compression Conference

[16] Nanrun Zhouab, Aidi Zhanga, Fen Zhenga, and Lihua Gongac, Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing, Optics & Laser Technology, volume 62 October 2014, Pages 152-160

[17] G. R. Jothilakshmi, R. J. Christilda, A. Raaza, Y. Sreenivasa Varma, V. Rajendran, "Extracting region of interest using distinct block processing method in sono-mammogram images", *2017 International Conference on Computer Communication and Signal Processing (ICCCSP)*, pp. 1-7, 2017.

[18] P. Praveenkumar, L. Kala, R. Nisha, K. Thenmozhi, J. B. B. Rayappan, R. Amirtharajan, "Hungarian sculptured Discrete Gould transformed bit planes-a double puzzled image", *2015 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-4, 2015.

[19] Ya-Pei Feng and Zhe-Ming Lu, an efficient hybrid feature for edge-preserving based on block truncation coding and tree-structured vector quantization with edge orientation classification of bit-planes, International Journal of Innovative Computing, Information and Control ICIC 2018 ISSN 1349-4198, Volume 14, Number 3, June 2018

[20] Jungrae Kim, Michael Sullivan, Esha Choukse, and Mattan Erez, Bit-plane compression: transforming data for better compression in many-core architectures, Proceeding ISCA '16 Proceedings of the 43rd International Symposium on Computer Architecture Pages 329-340

[21] Abdelatief H. Abouali,"BITPLANES BLOCK BASED LOSSY IMAGE COMPRESSION", International Journal of Engineering and Information Systems (IJEAIS),Vol. 2 Issue 10, OCTOBER – 2018, Pages: XX-XX

[22] Mehmet Utku, Gaurav Sharma, A. Murat Tekalp, "Gray-level-embedded lossless image compression" signal processing: image communication, Elsevier, volume 18, 2003, 443-454

[23] Ghadah Al-Khafaji, "image compression based on quad-tree and polynomial ", international Journal of computer applications, Volume 76 No. 3, August 2013.

[24] Rajasekhar Reddy , Ravichandran KS, Venkatraman B, Suganya SD," A new approach for the image compression to the medical images using PCASPIHT", Biomedical Research 2018; Special Issue: S481-S486

Dr. Abdellatief Hussien Abouali
Received B.Sc. from the MTC computer engineering 1984 presidential Honored, Master degree from collage of engineering Cairo University in expert systems, and received Ph.D. the University of Alabama in Huntsville (UAH) USA, Multi-class vector quantization for neural network design. Served in air force research in development till end of 2012, with many applied researches in areas of computers and computers based systems. Working for ElShrook Academy computer science department, Cairo, Egypt. Fields of interest image processing, neural networks, and machine learning

aabouali@hotmail.com , phone :+201120647588
aabouali4@gmail.com, dr.Abdullatif.Hussein@sha.edu.eg

APPENDIX A: Encoding Algorithm

Assume that given a square (or squared, null expanded) image matrix I of pixel resolution NxN, $N = 2^n$ (or null expanded to closest size satisfies the condition) with q bit colors/grays. Therefore, the image contains q bit-planes $(p_0, p_1, \ldots\ldots, p_{q-1})$ of size equal to I.

Definitions:

- ***Divide a square Block B of size $m > 2$, m is divisible by 2, that starts at location b_{sx}, b_{sy} as***

$DB(B, m, b_{sx}, b_{sy})$ *that produce block Set* $\{B1, B2, B3, B4\}$ ***each of size $m/2$ and their start locations are***

$(b_{sx}, b_{sy}), (b_{sx} + m/2, b_{sy}), (b_{sx}, b_{sy} + m/2), (b_{sx} + m/2, b_{sy} + m/2)$ ***consequently.***

- ***For a block B: B1C and BNNC are the block 1's count and None Null Count consequently.***
- ***Block B is said to be all zeros if block B1C=zeros.***
- ***Block B is said to be all ones if B1C=BNNC.***
- ***Encoding process of block B of size m that starts at location b_{sx}, b_{sy} as BENC(B, b_{sx}, b_{sy}, m) which output plane streams MBS and RBS.***

BENC (B, b_{sx}, b_{sy}, m)

{

If m=2

RBS=RBS+ row-scan (B)

else

{SWITCH (B1C (B), BNNC(B))

Case all zeros: MBS += '00'

Case all ones: MBS +='11'

Otherwise

MBS +='01',

$DB(B, m, b_{sx}, b_{sy}), \mathrm{BENC}(B1, b_{sx}, b_{sy}, m/2), \mathrm{BENC}(B2, b_{sx} + m/2, b_{sy}, m/2),$
$\mathrm{BENC}(B3, b_{sx}, b_{sy} + m/2, m/2), \mathrm{BENC}(B4, b_{sx} + m/2, b_{sy} + m/2, m/2).$

$\boldsymbol{MBS \mathrel{+}= MBS_1 + MBS_2 + MBS_3 + MBS_4}$
$\boldsymbol{RBS \mathrel{+}= RBS_1 + RBS_2 + RBS_3 + RBS_4}$

If size ((RBS + MBS) >= plane size) {RBS=row-scan of the plane; MBS=NULL; Return;}

}

Where MBS_i, RBS_1 are the main bit-stream and residual bit-stream of the sub-block i, i € {1,2,3,4}, the four sub-blocks of the main block out of the DB() function.

Basic Encoding(I)

{ MSB, RBS set to NULL for all planes.

Image I encoding

$= \boldsymbol{BENC}(p_0, \mathbf{0}, \mathbf{0}, N)$

$\cup\ \boldsymbol{BENC}(p_1, \mathbf{0}, \mathbf{0}, N) \cup \ldots\ldots\ldots.$

$\cup\ \boldsymbol{BENC}(p_{q-1}, \mathbf{0}, \mathbf{0}, N)$

}

Appendix B: Basic Decoding:

The encoded file header contains original image resolution that yields the number of bit-planes and the original image size. The size is squared and expanded to satisfy the former condition. Then a stack is initialized to recover planes through the decode block DECODEB (MBS, RBS).

DECODEB (MBS, RBS)

{

While (stack is not empty)

{

Popup $, b_{sx}, b_{sy}, m$

If Block intersection with original image matrix is ϕ then continue

Read from MBS stream two bits into tb

Case tb=00 set block to zeros

Case tb=11 set bock to ones

Case tb=01

If m==2

read from RBS four bits to set row wise block bits.

else

$$DB(B, m, b_{sx}, b_{sy}), \mathrm{push}\,(\mathrm{b_{sx}} + \mathrm{m/2}, \mathrm{b_{sy}} + \mathrm{m/2}, \mathrm{m/2}),$$
$$\mathrm{push}\,(\mathrm{b_{sx}}, \mathrm{b_{sy}} + \mathrm{m/2}, \mathrm{m/2}), \mathrm{push}\,(\mathrm{b_{sx}} + \mathrm{m/2}, \mathrm{b_{sy}}, \mathrm{m/2}), \mathrm{push}\,(\mathrm{b_{sx}}, \mathrm{b_{sy}}, \mathrm{m/2}).$$

}

}

The decode procedure has two binary streams MBS, RBS

and is as following: -

If MBS is NULL

row bit-set from RBS

else

{

$$\mathrm{stack\,push}\,(0,0,\mathrm{N}).$$

DECODEB (MBS, RBS).

}