International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 2 Issue 10, October – 2018, Pages: 18-26

Bitplanes Block Based Exact Image Compression

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

Abstract: *In this paper, an exact image compression based on bit-planes blocking is proposed. The proposed algorithm uses two bit codes for block representation. The codes represent the states of Unicode block and non-Unicode. The algorithm considers further division to non-Unicode block. The block division continues until the smallest block size which are kept as residuals. The smallest block size in the study is two by two. The main process of encoding consumed three codes. Subsequent process uses the fourth code for further compression. The resultant file is subject to further exact compression. The compression technique considered in this study is Huffman. The compression-decompression implementation complexity is comparable with the well-known methods. Also, the compression ratio for the algorithm is comparable with well-known methods. The algorithm parallelization is straightforward and dependent on number of planes. Within a plane, the process hardware realization is simple and does on require special hardware.*

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

1. INTRODUCTION

Multi-media is a combination of contents like audio, text, video, digital images, graphics, as well as interactive interfaces. Multi-media uses and application domains are growing very fast. The success of multimedia is highly tightened with effective encoding to components of high memory demands such as videos and digital images. Demands and the volume of digital images used in: 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. Researchers, in our opinion, should consider development of special image memory and/or image processors or at least a new generation of graphic processors to be of more balanced dual functions for both graphics and image processing. To date, human effective storage, processing, recognition, indexing, and recall is far above all developed methodologies and devices man made.

Image encoding/decoding is a vital process in success of computer vision systems that intensively use images [2-4]. Image encoding basically is an effective representation to digital images. The encoding process, in computer vision systems, increases systems ability 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 [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 data. 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]. Attributes of good encoding scheme includes: 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 reality of images which is 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].

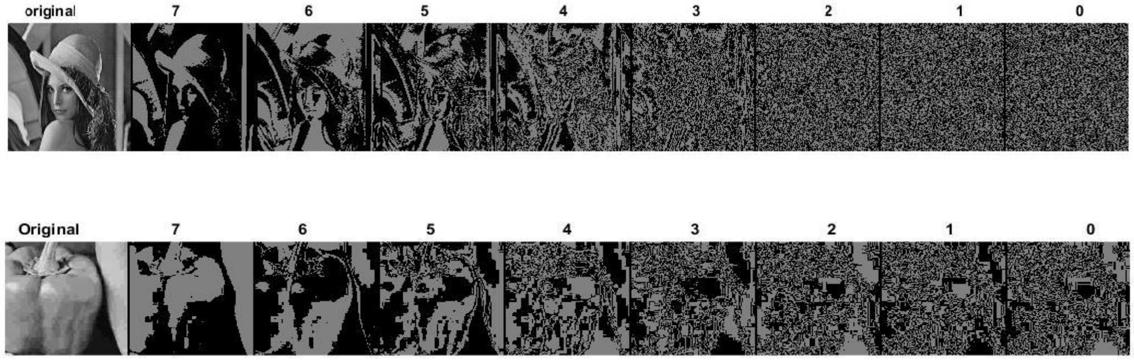

Figure (1) Lena, pepper original and bit-planes for 8-bit gray

In this paper, an 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. An optional additional exact encoding step is possible for the sake of completion Huffman encoding is considered in the study.

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

2. THE PROPOSED ENCODING/DECODING ALGORITHMS

The basic encoding algorithm is a three-code encoding algorithm. The three-code algorithm uses two-bit codes 11,00, and 01. These codes represent cases of a block all zeros, all ones and a mix, consequently. The outcome of the algorithm is two-bit stream. The first stream represents how blocks are divided and where are the blocks of unary codes which we call Main Bit Stream, MBS. The second is four-bit Residual Bit Stream, RBS, contains row-scan of the smallest blocks, two by two, of non-unary bit values. The fourth code is used for further optimization to the MBS using Adaptive Run Length Encoding ARLE.

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',

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

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

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

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.

$$\boldsymbol{Image}\ I\ \boldsymbol{encoding} = \boldsymbol{BENC}(p_0, \mathbf{0}, \mathbf{0}, N) \cup \boldsymbol{BENC}(p_1, \mathbf{0}, \mathbf{0}, N) \cup \ldots\ldots\ldots \cup \boldsymbol{BENC}(p_{q-1}, \mathbf{0}, \mathbf{0}, N)$$

}

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

ISSN: 2000-000X
Vol. 2 Issue 10, October – 2018, Pages: 18-26

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

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

}

}

The decode procedure has two binary streams MBS, RBS and is as following:-

If MBS is NULL

row bit-set from RBS

else

{

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

DECODEB (MBS, RBS).

}

3. EXPERIMINTAL RESULTS, ADDETIVE ENCODING AND DISCSSION

Our first experiment uses the raw algorithm, described above, against a set of standard images. The used set was aerial, cameraman, woman-house, Barbara, Lena, and house, Figure (2). The colored ones changed to gray using ‘Matlab’ ‘rgb2gray’ function. Table (1) summarizes the results. The results show that the encoding yielded little better results compared to Huffman encoding in all except for aerial. Table (2) shows the compression ratios per plane, skipping the size limitation for sake of the study. The table shows better compression ratios for Most Significant Bit Planes, MSBP. That comes from the fat probable blocking is more probable. While the Least Significant Bit Planes, LSBP, compression yields expansion.


Aerial


Cameraman


Woman-House


Barbara


Lena


House

Figure (2) set of images used in the study

Table (1) Basic Algorithm

	Huffman	Entropy	Proposed Basic
aerial	1.11	7.1781	1.04
Cameraman	1.10	7.2074	1.239
Woman House	1.05	7.5598	1.261
Barbara	1.04	7.6321	1.129
Lena	1.05	7.5691	1.181

House	1.23	6.4971	1.212

Table (2) Bit Planes Compression Ratios

	Aerial	Cameraman	Woman House	Barbara	Lena	House
Plane 0	0.648	0.654	0.700	0.649	0.648	0.648
Plane 1	0.648	0.659	0.764	0.649	0.646	0.659
Plane 2	0.647	0.687	0.866	0.655	0.657	0.703
Plane 3	0.654	0.790	1.005	0.707	0.718	0.913
Plane 4	0.679	0.992	1.167	0.825	0.881	1.171
Plane 5	0.773	1.414	1.450	1.018	1.122	1.427
Plane 6	0.982	1.957	2.035	1.459	1.696	1.686
Plane 7	1.505	4.187	3.269	2.374	3.392	2.211

The samples of the MBS codes for the 8 planes of Barbara are in table (3). The table indicates that MSBP contains more unary-code blocks than LSBP. From the table, also, the consecutive existence of the '01' codes are eminent. This was the motive behind the second step, ARLE.

Table (3) Barbara Planes Structural codes

	Structural Codes
Plane 0	0101010101010101010101010101010101010101010101010101010101010101
Plane 1	0101010101010101010101010101010001010101010101010101010101010101
Plane 2	0101011101010101010101010101010001010111010101010101010101110101
Plane 3	0101010101010101010101010001010101010101010101010101000101010101
Plane 4	0101010101010101010101000101000101110101010001010100011101010101
Plane 5	0101010101010101010101000101000101110101010000010100000001010101
Plane 6	0101010101010101010111010111010100010101111101011111110101010101
Plane 7	0101010101011111010111110101011111010111110101010101110101011111

Adaptive Run Length Encoding

Considering the unused code, '10', and the concept of fixed Run Length Encoding, RLE for further compression. Figure (3) shows the use of fixed RLE of length 1,2,3,4,6,7 against the MBS compression ratio on camera-man image. In fact, this figure was found typical for all other images with minor changes. The figure indicators are: MSBP are less in compression ratio and low in run length and vice versa for LSBP. The ARLE used to code a fixed run length, per plane, of '01' codes. The run length, L, is bounded by: $2 >= L <= (\log_2(N) - 1)$. Table (4) show the compression ratios after ARLE compared to Huffman.

Block Coding

Dividing image plane into blocks then encoding each block, on one side set an upper limit to unary block size, which is encoded in only two bits, as the block size and on the other hand reduces successive block split codes as the size of the block is smaller case of existence to small unary blocks is far more probable than large ones. Figure (4) shows the effect of blocking

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 2 Issue 10, October – 2018, Pages: 18-26

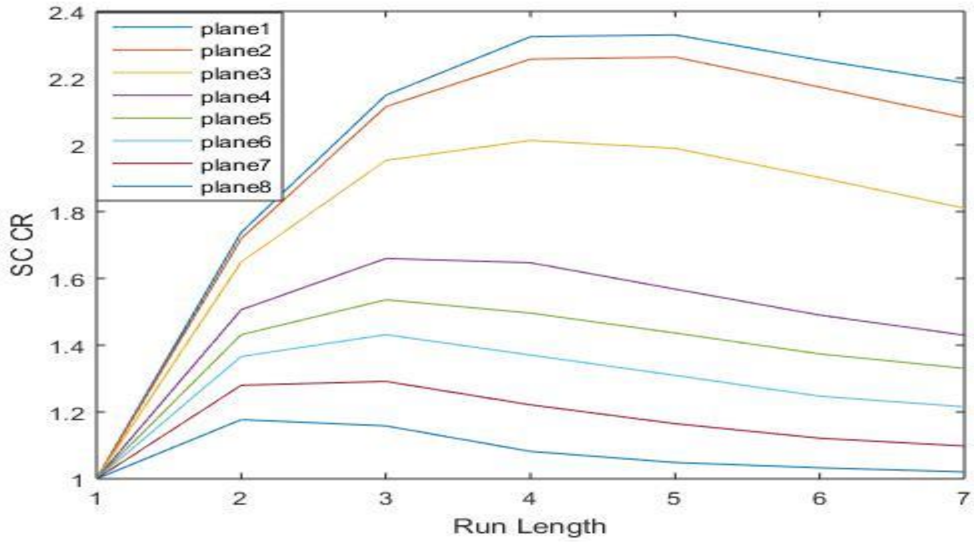

Figure (3): Run Length effect on compression ratio for Camera-Man Planes

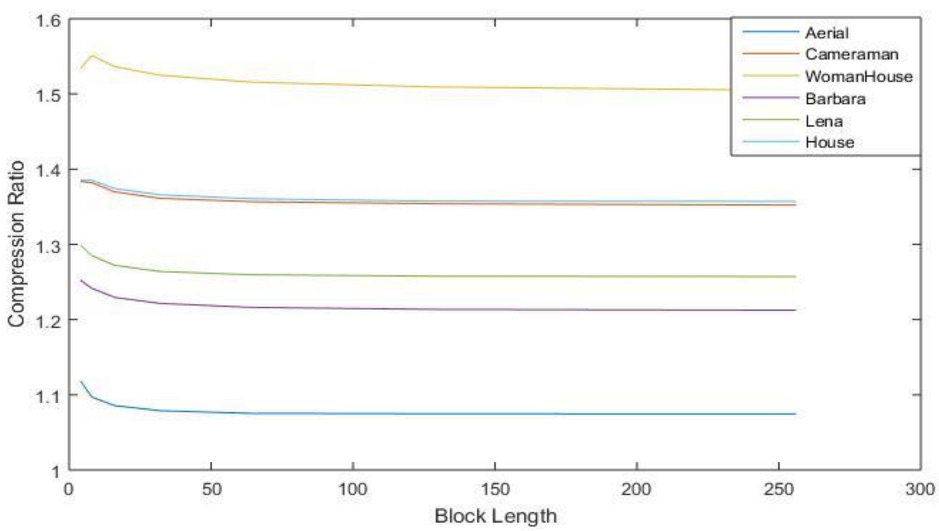

Figure (4): Block Size (4,8,16,32,64,128,256) Compression Ratio

The block sizes used in the experiment are: 4x4, 8x8, 16x16,32x32, 64x64,128x128,256x256. Better compression ratios are towards smaller block sizes. However, from the figure one can easily infer that the optimal is not the smallest. That comes from the fact if the images contain too much details its MBS will contains too much split codes, ‘01’, while the size of RBS is same. One can easily infer that the case could be completely different if the nature of the images is different.

Huffman Encoding

The residual stream bytes do not contain codes: ‘0000xxxx’, ‘1111xxxx’, ‘xxxx0000’, and ‘xxxx1111’ which are 60 codes. Also, the main streams bytes could contain patterns that comes from successive split or zero’s/one’s regions. That as well as the planes that are kept as is due to the non-blocking natures. These leads to the possibility of adding more compression step using one of the exact compression techniques such as LZW, arithmetic, and Huffman for sake of more compression. In our study. we tried Huffman Encoding. Table (4) shows the compression ratios for the test images ARLE, ARLE with Huffman against standard Huffman.

Table (4) Shows the compression ratios for the proposed with adaptive RLE

	Huffman	Proposed +ARLE	Proposed +ALRE +Huffman

aerial	1.11	1.05	1.191
Cameraman	1.10	1.27	1.456
Woman House	1.05	1.42	1.524
Barbara	1.04	1.14	1.316
Lena	1.05	1.187	1.353
House	1.23	1.289	1.478

LAB SET KIT

Figure (5) block oriented images

Discussion

The presented algorithm offers an exact compression technique with significant improvement compared to Huffman. One can easily infer that the nature of standard images does not coincide with the proposed algorithm nature as they are selected to contain significant details of wide nature. To make up our point the figure (5) contains the other side type of images that contains significant unary-code blocks. Table (5) contains the results compared to Huffman.

The algorithm used performance with RGB color images will not differ from performance on gray dealing with them as three separate matrices. Tables (6) contains the results on the two color images, Woman house and Aerial, in our set. The results show performance similar to the gray case.

The low pass filters, in general, reduces the spark noise. The spark noise affects the blocking. So, applying low pass filters on images before compression affects the compression ability of the proposed algorithm. Table (7) presents a comparison between compression before and after 3x3 median filter.

A general note on all results the performance of the algorithm for Aerial image is significant low compared to others. The reason behind that degradation in results is there are too much spatial details exits of few pixels. As well as these details are skewed relative to the image coordinates.

Also, employing these techniques for encoding successive fames of videos after subtraction could lead to significant compression. An adapted motion prediction like encoding coincide with the algorithm encoding. Also, one can see that the algorithm could be easily adapted to lossy encoding.

Table (5) Compression ratios for Lab-set and Kit images

	Proposed	Huffman
LAB SET	21	1.65
KIT	5	2.3375

Table (6) Color Images Encoding

	Huffman	Basic	+ARLE	+Huffman
Aerial	1.10	1.05	1.12	1.21
Woman House	1.08	1.26	1.40	1.52

Table (7) |Effect of median filter on compression

	Before filter	After median filter
Cameraman	1.456	1.6656
Barbara	1.316	1.4571

4. CONCLUSION

A block based exact image compression is proposed. The proposed algorithm used basically three codes, two bit, to encode a block. The output of the basic algorithm is two streams MBS, and RBS. The MBS contains the splitting codes, and the fill codes of unary code blocks of size greater or equal to two. The RBS contains a row scan for blocks of size 2x2 that are of non-unary code. The fourth of the two-bit code is used for an ARLE compression. The encoding process is simple and its hardware realization is possible. The MSB planes satisfies compression ratios higher compared to LSB's. The algorithm could be used block based. Small block sizes show compression ratios higher for images with too much details. The performance of the algorithm is significant better compared to Huffman for even the well-known standard images. The performance on RGB images is similar to that of gray images.

The adaption of the algorithm to video encoding is promising. The algorithm could be extended to provide several adaptive lossy techniques. Allowing losses could be increased as you move toward LSBP's. The loss could be in setting a percent for assuming unary blocks, removing RBS from LSBP's, as well as combination of the two.

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

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 of classified 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