International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359

Research Article | **January 2013**

An Improved JPEG Algorithm Based on Psycho visual Redundancy and Coding Redundancy

Md. Akkas Ali*
Lecturer, Dept. of CSE & IT,
University of Information Technology and Sciences
Dhaka-1212, Bangladesh

Md. Golam Kibria
Asst Prof & Head, Dept. of CSE & IT,
University of Information Technology and Sciences
Dhaka-1212, Bangladesh

Md. Mehedi Hasan
Software Engineer,Voxvalley Private Technologies Limited.
Dhaka-1212, Bangladesh

Md. N Bhuiyan
Lecturer,
Dept. of CSE, University of Information Tech. and Sciences, Dhaka-1212, BD

Jahangir Alam S.M.
Dept. of Mechanical & Electrical Engineering
Xiamen University,
Fujian Province 361005, P.R. China

Abstract—

In computer science and information theory, data compression and source coding is the process of encoding information using fewer bits. Compression is useful because it helps to reduce the consumption of expensive resources such as hard disk space or transmission bandwidth. For digital image, JPEG is one of the most worldwide used image compression methods. This method is able to get good results with moderate compression but not provides better compression at different bit rates. For this we have proposed a algorithm with lossy technique that improves the image quality of standard JPEG at different bit rates while keep compatibility with the baseline JPEG. In our method we can get more compression by discarding less important frequencies means high frequencies that human eye will not detected. To get better compression ratio we reduce the coding redundancy in standard JPEG. The experimental result shows that in average 0.25 bpp the proposed algorithm provides 3.61 percent better PSNR and in average 0.5 bpp the proposed algorithm provides 10.24 percent better PSNR than the conventional JPEG algorithm.

Keywords*— *Image compression, image quality management, PSNR, lossy technique, JPEG algorithm

I. INTRODUCTION

The memory capacities of computers have increased as new technologies are emerging; the requirement for more storage space is also increasing as more data are needed to be stored. In the case of image data, the spatial and color resolutions are increased for the betterment of image quality, thus it requires more space to store images and more time while sending over network. Image compression minimizes the size of image without degrading the quality of image [1]. Image compression is an application of data compression that encodes the original image with few bits. The objective of image compression is to reduce the redundancy of the image and to store or transmit data in an efficient form. The main goal of such system is to reduce the storage quantity as much as possible, and the decoded image displayed in the monitor can be similar to the original image as much as can be. The underlying basis of the reduction process is the removal of redundant data [2]. From a mathematical point of view this amount to transformation of 2-D pixel array into a statistically uncorrelated dataset. The transformation is applied prior to storage or transmission of the image. At some later time, the compressed image is decompressed to reconstruct the original image or an approximation to it.

II. PROPOSED MODEL FOR IMAGE COMPRESSION

The lossy sequential DCT-based mode is popular model for image compression. It gives good PSNR values and quite better compression ratio for every image. To achieve higher compression ratio and quality here we develop the lossy sequential DCT-based mode for gray scale image [3]. We modify the conventional JPEG algorithm in quantization level and also modify JPEG default Huffman table for encoded image [4]. We try to discard the high frequency AC coefficients that human eye will not detected because human eye is fairly good at seeing small differences in brightness over a relatively large area. In JPEG the high frequency AC coefficients discarded is done by simply dividing each coefficient by a constant for that component, and then rounding to the nearest integer. But this is not properly done for every high frequency AC coefficients. To do it we apply our proposed algorithm after rounding the nearest integer in quantization step.

1. The image is broken into 8*8 blocks of pixels,
2. Working from left to right, top to bottom, the DCT is applied to each block,
3. Each block is compressed through quantization,
4. Discarding high frequency AC coefficient values coming after DC coefficient or non-zero AC coefficient values,
5.Zigzag order is applied in each block,

6. DC coefficients are encoding through DPCM and AC coefficients are encoding through run length coding, 7. Finally sequence of bit stream is stored that reduced the amount of space for image [5].

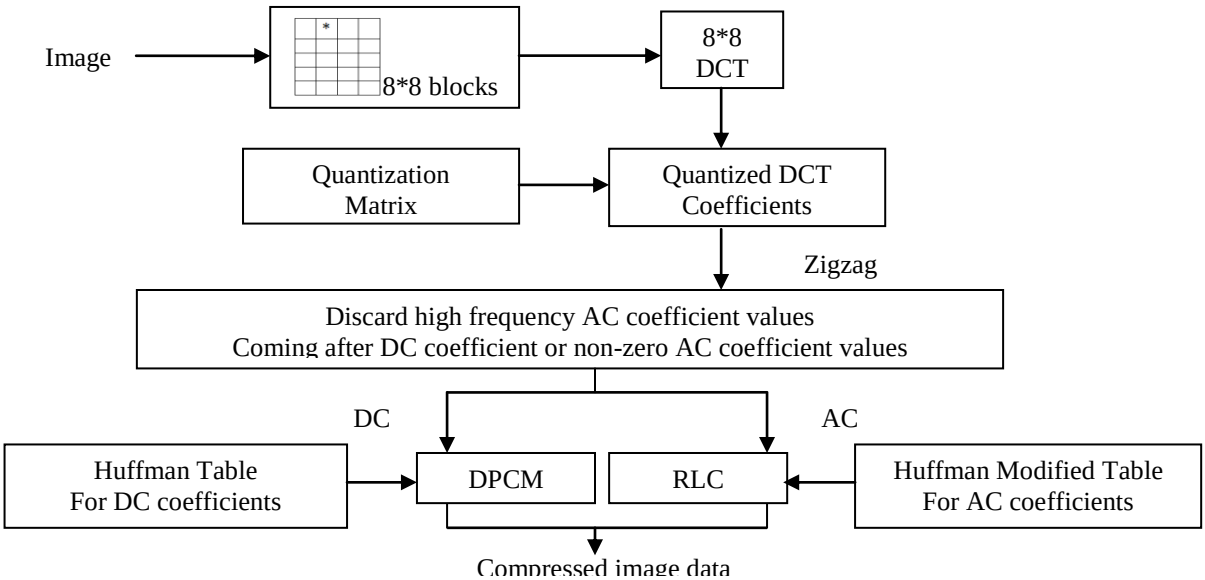

Fig. 1 Proposed algorithm for image Compression

III. PROPOSED MODEL FOR IMAGE DECOMPRESSION

After encoding the image we get the sequence of bit stream that is the compressed image data. If we see the compressed image the reconstruction is must be needed. The image decoding process starts with reconstruction of the DC and AC coefficients based on the given Huffman table [6]. This is followed by a multiplication of all coefficients with the corresponding scaling factor Q. The model for JPEG image decompression is shown in fig. 2.

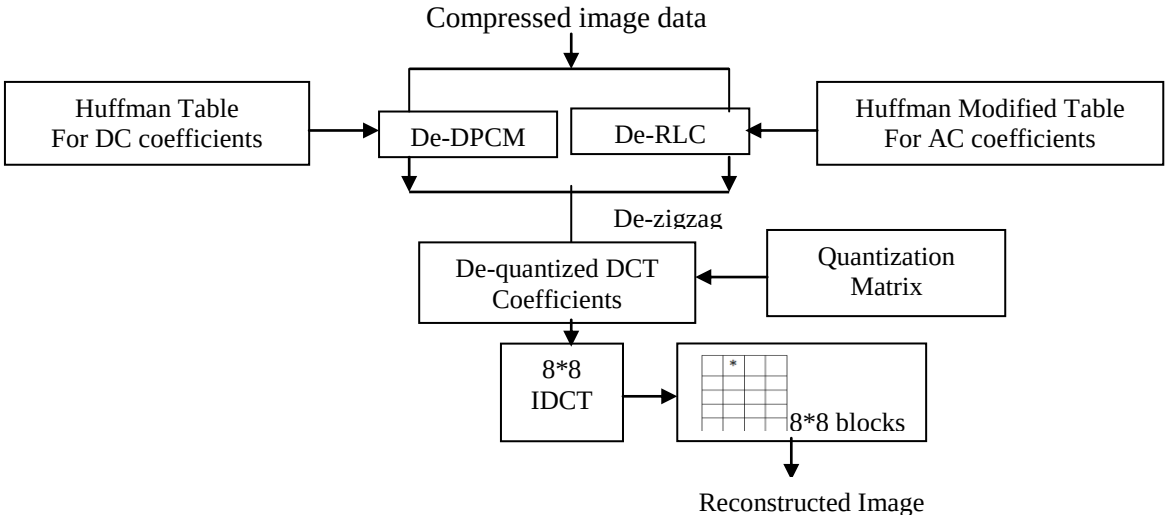

Fig. 2 Algorithm for image Decompression

IV. IMPLEMENTATION OF OUR PROJECT WORK

A. Image Preparation

For implementation of our project work here we described how the image pixel values are compressed to our methodology. According to the methodology we take a gray scale image as an input, broken the image into 8*8 blocks of pixels [7]. If we take 512 * 512 matrix of image the total image block is 4096. But we have taken one block for our implementation work.

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

B. Level Shift process

The DCT is designed to work on pixel values ranging from -128 to 127, the original block is "level off" by subtracting 128 from each entry. This results in the following matrix.

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

C. Discrete Cosine Transform

The DCT matrix is:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

The matrix is ready to perform the discrete cosine transform which is accomplished by matrix multiplication.

D = TMT' (1)

In equation matrix M is first multiplied on the left by the DCT matrix T from the previous section, this transforms the rows. The columns are then transformed by multiplying on the right by the transpose of the DCT matrix. This yields the following matrix.

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

D. Quantized DCT Coefficients

The 8*8 block of DCT coefficients is ready for compression by quantization. Quantization is achieved by dividing each element in the transformed matrix D by the corresponding element in the quantization matrix, and then rounding to the nearest integer value [8]. For the following, quantization matrix Q_{50} is used.

$$C_{i,j} = \text{round}(D_{i,j} / Q_{i,j}) \quad (2)$$

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

E. Zigzag Scan

After using the zigzag order we obtain one dimensional vectors in the following sequence:

10, 4, 3, -7, 9, 2, 5, 1, -5, -3, -2, -5, 1, 2, 1, 0, 1, -2, 0, 1, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0 .....

F. Discard High Frequency AC Coefficient Values Coming after DC coefficient or none Zero AC Coefficient Values

After zigzag scan the AC coefficient zero values represent that the pixel values of the corresponding block usually have lower variance. That is, the pixel values of the block will be close to their mean value. In most cases, not all high

frequency AC coefficients are equal to zero [9]. For this, we try to cancel those high frequency AC coefficient values using our algorithm. Now the bit sequence is:

10, 4, 3, -7, 9, 2, 5, 0, -5, -3, -2, -5, 0, 2, 0, 0, 1, -2, 0, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0 …...

G. Assign the End of Block Symbol

After finishing algorithm here we assign the end of block symbol then the sequence is:

10, 4, 3, -7, 9, 2, 5, 0, -5, -3, -2, -5, 0, 2, 0, 0, 1, -2, 0, 1, 0, 0, 0, 0, -1, EOB

H. DC Differential Coding

This process allows the first DC value of the first block to be passed through to the entropy coding module. Afterward, the value passed is a difference between the current block DC value and the previous block. Here we get only first block, so according to the procedure DC is encoded by the following bit technique:

DC = 10
Category = 4
Base code = 101
Category code for 10 is = 1010
So, for DC value 10 the bit stream is: 1011010

I. Run Length Coding for AC Coefficients

AC coefficients are encoded by the use of run length coding technique, are described in following. The AC sequence is: 4, 3, -7, 9, 2, 5, 0, -5, -3, -2, -5, 0, 2, 0, 0, 1, -2, 0, 1, 0, 0, 0, 0, -1, EOB

First count zero and put a pattern like this: (0,4) ; (0,3) ; (0,-7) ; (0,9) ; (0,2) ; (0,5) ; (1,-5) ; (0,-3) ; (0,-2) ; (0,-5) ; (1,2) ; (2,1) ; (0,-2) ; (1,1) ; (4,-1) ; EOB

Then the use of default Huffman table for AC coefficients and category table we put a pattern like this:

(0/3,4) ; (0/2,3) ; (0/3,-7) ; (0/4,9) ; (0/2,2) ; (0/3,5) ; (1/3,-5) ; (0/2,-3) ; (0/2,-2) ; (0/3,-5) ; (1/2,2) ; (2/1,1) ; (0/2,-2) ; (1/1,1) ; (4/1,-1) ; EOB

Now use the bit sequence of default Huffman table for AC coefficients and category table we obtain bit stream like this: 100100 0111 100000 10111001 0110 100101 1111001010 0100 0101 100010 11100110 110111 0101 11001 1110110 00

So the final bit stream we obtain is: 1011010 100100 0111 100000 10111001 0110 100101 1111001010 0100 0101 100010 11100110 110111 0101 11001 1110110 00

It is 97 bits. The whole 8*8 block has 512 bits. It gives compression ratio is: 512/97 = 5.27: 1 and the 8*8 block has 64 pixel values, so the bit/pixel is: 97/64 =1.51 bit/pixel

V. EXPERIMENTAL RESULTS

A. Image Quality Measurement

Measurement of the quality of image compression is important for image processing application. Now a data compression is important for storage and transmission [10]. In general, measurement of image quality usually can be classified into two categories

1. Subjective quality measurement, 2. Objective quality measurement

A.1. Subjective Quality Measurement

Subjective quality measurement depends on human visual system. If image is looking good on human eyes then image quality is subjectively better.

A.2. Objective Quality Measurement

In objective quality measurement MSE and PSNR are the most common measures of image quality in image compression systems. The large value of MSE means that image is poor quality. The small value of Peak Signal to Noise Ratio (PSNR) means that image is poor quality. MSE and PSNR are defined as follow: PSNR = 10 log (2552 / MSE) (3)

B. Comparative Analysis

We have proposed a new lossy technique algorithm with JPEG default Huffman table modification that improves the image quality of the JPEG standard for different bit rates while keep capability with the baseline JPEG [11]. The PSNR values obtained in the JPEG encoding of six typical images at 0.5 and 0.25 bits per pixel using our algorithm and the JPEG standard algorithm are shown in table 1. Our experimental result shows that in average 0.25 bpp the proposed algorithm provides 3.61 percent better PSNR and in average 0.5 bpp the proposed algorithm provides 10.24 percent better PSNR than the conventional JPEG algorithm.

C. Quality Analysis

To quality analysis in the image here we took two single image pepper and bird. Our algorithm provides better PSNR values than traditional JPEG algorithm at different bit rates [12]. The objective analysis at different bit rates on pepper image shown in table 2. In subjective quality tests, image testers select our method as a better image when the difference between both images was appreciable by human eyes.

TABLE I
COMPARATIVE ANALYSIS OF DIFFERENT IMAGES, AT BITRATES: 0.25 BPP VS 0.5 BPP

IMAGE NAME	JPEG(PSNR)		OUR (PSNR)	
	0.25 bpp	0.5 bpp	0.25 bpp	0.5 bpp
Pepper (512 * 512)	15.51	24.09	16.55	28.09
Barbara (512 * 512)	14.55	17.23	14.93	18.26
Bridge (512 * 512)	14.06	15.60	14.31	16.11
Air plane (512 * 512)	10.81	22.83	11.72	33.47
Bird (768 * 512)	18.33	27.80 (0.40 bpp)	20.37	36.01 (0.40 bpp)
Tiger (512 * 512)	12.88	15.51	13.15	16.67
Lake house (512 * 480)	17.36	21.51	17.98	23.28
Murti (376 * 480)	13.12	17.99	13.79	20.49
Pentagon (376 * 376)	19.33	23.12	19.60	23.72
Baboon (512 * 512)	16.32	17.54	16.52	17.92

TABLE II
QUALITY ANALYSIS OF JPEG AND OUR ALGORITHM FOR PEPPER IMAGE

IMAGE NAME	JPEG(PSNR)		OUR (PSNR)	
	0.25 bpp	0.5 bpp	0.25 bpp	0.5 bpp
Pepper (512 * 512)	15.51	24.09	16.55	28.09
Barbara (512 * 512)	14.55	17.23	14.93	18.26
Bridge (512 * 512)	14.06	15.60	14.31	16.11
Air plane (512 * 512)	10.81	22.83	11.72	33.47
Bird (768 * 512)	18.33	27.80 (0.40 bpp)	20.37	36.01 (0.40 bpp)
Tiger (512 * 512)	12.88	15.51	13.15	16.67
Lake house (512 * 480)	17.36	21.51	17.98	23.28
Murti (376 * 480)	13.12	17.99	13.79	20.49
Pentagon (376 * 376)	19.33	23.12	19.60	23.72
Baboon (512 * 512)	16.32	17.54	16.52	17.92

This test shows that our method gets better PSNR for the same bit rate in pepper image. The shape curve for both methods is similar, but our algorithm is always better in standard JPEG [13]. In subjective quality tests, image testers select our method as a better image when the difference between both images was appreciable by human eyes.

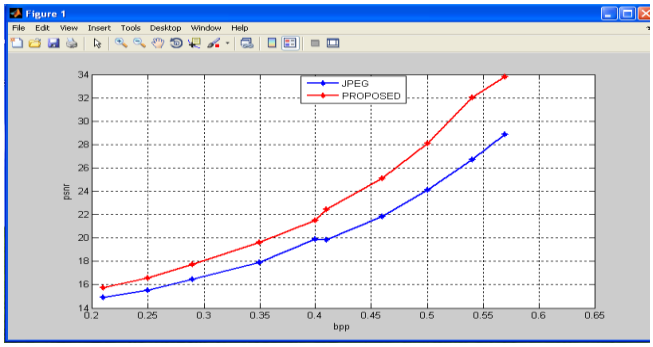

Fig. 3 Pepper image PSNR vs. bpp results

Fig. 4 Reconstructed pepper image using our algorithm (0.40 bpp)

TABLE III
QUALITY ANALYSIS OF JPEG AND OUR ALGORITHM FOR BIRD IMAGE

Bits/pixel (bpp)	Compression Ratio	PSNR (JPEG)	PSNR (OUR)
0.4026	19.86 : 1	27.8025	36.0125
0.3755	21.30 : 1	24.9687	32.1458
0.3062	26.12 : 1	20.3749	23.1271
0.2559	31.25 : 1	18.3307	20.3750
0.2074	38.55 : 1	16.8482	18.2818

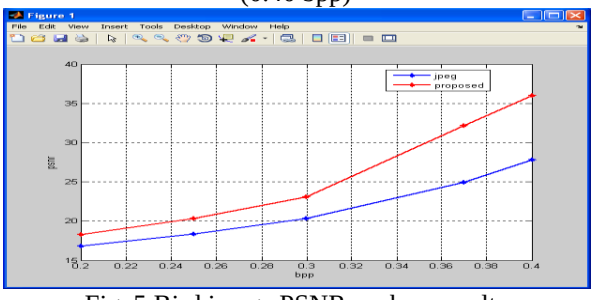

Fig. 5 Bird image PSNR vs. bpp results

Fig. 6 Reconstructed bird image using our algorithm (0.25 bpp)

Simulation results indicate that our proposed system works well enough. Our algorithm always provides better PSNR results. It also provided better result at different bit rates. To test subjectively our algorithm, we asked the opinion of several people with respect to the quality of the images generated by our algorithm and by using standard JPEG, our algorithm always provided better scores. At quality factor 50 our Algorithm not provides highest PSNR that standard JPEG provided. But at quality factor 50 our algorithm provides highest compression ratio that standard JPEG not provided.

REFERENCES

[1] Wei–Yi Wei, "*An introduction to image compression*", National Taiwan University.
[2] JPEG, "*Joint Photographic Expert Group, JPEG Home Page*", Retrieved from *http://www.jpeg.com.*
[3] Gregoty K. Wallace, "*JPEG still compression standard*", commun of the ACM, 34(4), 1991, 31-44.
[4] S.W. WU and A. Gersho, "*Rate-constrained picture adaptive quantization for JPEG baseline coders*", IEEE ICASSP-93, vol. V, pp. 390-392.
[5] Julio and Josep Prades, "*Adaptive quantization with JPEG coding*", University of Politecnica De Valencia.
[6] Golomb-Rice, "*Adaptive coding of DCT coefficients*", IEEE Transaction and Information Theory.
[7] Edmund Y. Lam and Joseph W. Goodman, "*A mathematical analysis of the DCT coefficient distribution for images*", IEEE Transaction and Information Theory
[8] Ken Cabeen and Peter Gent, "*Image compression and discrete cosine transform*", College of the Redwoods
[9] H.S. Malvar and D.H. Staelin, *The LOT: Transform coding without blocking effects*, IEEE Transaction on Acoustic, Speech and Signal Processing, 37(4), 1989, 553-559.
[10] R. Rosenholtz and A.B. Watson, "*Perceptual adaptive JPEG coding"*, Proc. IEEE International Conf. on Image Processing, Laussanne, Switzerland, 1996, Vol. Ipp. 901-904.
[11] Information technology: "*Digital compression and coding of continuous-tone still images extensions*". CCITT Recommendation T.84, 1996.
[12] W.B. Pennebaker, J.K. Mitchell, JPEG: "*Still Image Data Compression Standard",* (New York: Van Nostran Reinhold, 1993).
[13] "*Digital Compression and Coding of Continuous Tone Image*" (Part 1: Requirements and Guidelines), ISO/IEC 10918-1, 1992.