IMPROVE IMAGE COMPRESSION USING THE LIFTING SCHEME AND THE DISCRETE COSINE TRANSFORM

Walaa M. Abd-Elhafiez[1], Wajeb Gharibi[2]
[1]Computer Science Department, Faculty of Science, Sohag University, 82524, Sohag, Egypt.
[1,2]Faculty of computer Science and Information Systems, Jazan University, Jazan, KSA.
Corresponding Auther Email: walaa.hussien@science.sohag.edu.eg
[2]Email: gharibi@jazanu.edu.sa

ABSTRACT: *In this paper, adaptive and hybrid method is proposed to increase the compression ratio for the gray images using the lifting transform (LT) and the discrete cosine transform (DCT). Lifting transform is used to reduce the image size without losing much of the resolution computed. Also, the discrete cosine transform is a fast transform and has fixed basis images that give good compromise between information packing ability and computational complexity. The propose algorithm for image compression utilizing the advantages of both transforms. Experimental results show that the coding performance can be significantly improved using the proposed scheme compared to existing methods as the JPEG-based DCT, the JPEG-2000 and the set partitioning in hierarchical trees (SPIHT) coder, which the hybrid DCT transform with lifting transform can reduce transmission cost to memory space to store the encoded data and transmit the encoded data.*

Keywords: Image compression, Lifting transform, Q-coder, Quantization, Discrete cosine transform.

INTRODUCTION

Image compression has become the most recent emerging trend throughout the world. The objective of image compression is reduce the redundancy of the image and to store or transmit [1, 2, 3] data in an efficient form. Image compression is the process of encoding image such that less storage space is required to archive them and less transmission time is required to retrieve them over a network. Compression is possible because most images contain large sections (e.g. backgrounds) that are often smooth, containing nearly identical pixel values that contain duplicate information. This is referred to as ‘statistical redundancy’. Ideally, an image compression technique strives to remove redundant information, and efficiently encoded and preserve the remaining data. Some of the common advantages image compression over the internet are reduction in time of webpage uploading, downloading and lesser storage space in terms of bandwidth. Compressed images also make it possible to view more images also make it possible to view more images in a shorter period of time [4]. Image compression is essential where images need to be stored, transmitted or viewed quickly and efficiently.

Image compression algorithm may be broadly categorized into two types: lossy and lossless [5, 6, 7]. With lossy image compression, redundant pixel data are discarded during the compression process so that the compressed image is only an approximation of the original material. Quite often adjusting the compression parameters can vary the degree of lossless allowing the image-maker to trade off file size against image quality. In lossless compression schemes, the reconstructed image (after compression) is numerically identical to the original image and may be displayed as an exact digital replica of the original. Only the statistical redundancy is exploited to achieve compression, in general, lossless techniques provide far lower compression ratios than lossy techniques with the bouns of preserving all image content.

Sachin D. [8] have reviewed and summarized the characteristics of image compression, need of compression, principles behind compression, different classes of compression techniques and various image compression algorithms based on Wavelet, JPEG, vector quantization, and Fractal approaches. Janaki R. and Tamilarasi [9] proposed a technique for image compression which used the wavelet-base image coding in combination with huffman encoder for further compression. It aims to determine the best threshold to compress the still image at a particular decomposition level by combining the EZW encoder with huffman encoder. H. Singh and S. Sharma [10] have presented a hybrid DWT, DCT and huffman algorithm for image compression. In [11], B. Sriram and S. Thiyagarajan have applied similar hybrid algorithm but reduces blocking artifacts, ringing effects and false contouring appreciably. Hao et. al [12] proposed a directional DCT- like transform basing on the lifting structure of DCT. The transform can be performed along arbitrary direction in theory and still has the perfect reconstruction property. Also, JPEG-wise image coding scheme evaluate the performance of the proposed directional DCT- like transform is proposed, which each block of size 8×8 is transformed along variable direction in order to better adapt to local orientation properties of image. Wenbing et. al [13] presented the improved algorithm according to image compression technology tp pledge the real time of the image transmission and gain the high compression ratio under the image quality. The improved SPIHT image coding algorithm based on fast lifting wavelet transform presents fast lifting wavelet transform to improve transform course, because of many consecutive zero appearing in SPIHT quantification coding, adopting the simultaneous encoding of entropy and SPIHT. Entropy coding adopts run-length changeable coding.

In this paper, an efficient method using lifting transform, discrete cosine transform and Q-coder technique for image compression is presented, which divide the entry image into 16×16 blocks, the one level 2-D lifting transform [14] is taken on each block, then applying the 2-D DCT followed by applying Q-coder.

LIFTING TRANSFORM

The lifting scheme [14, 15] has many important advantages, including in place operation, perfect reconstruction capability, simplicity in implementation and parallelism in computing. It has three kind of operations, as shown in figure 1, Split, Predict and Update. In spilt step the original data is divided into two disjoint subsets (even and odd indexed samples). The objective of predict step is compute a prediction for the odd samples, based on the even samples (or vice versa). This prediction is subtracted from the odd samples creating an error signal. The operation of obtaining the differences from the prediction is called as a lifting step. The update step follows the prediction step, where the even values are updated from the input even samples and the updated odd samples. They become the scaling coefficients, which will be passed on to the next stage of the transform. This is second lifting step. The forward and inverse lifting transform is shown in figure 1(a) and 1(b).

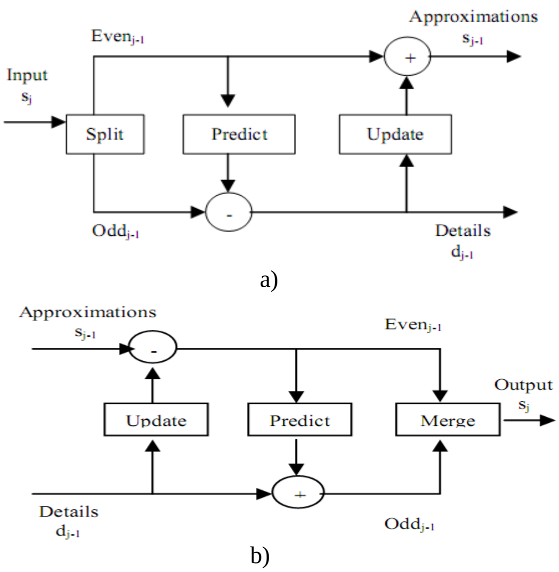

Fig. 1 Block diagram of lifting transform a) Forward lifting transform and b) Inverse lifting transform

The odd elements are replaced by the differences and the even elements by the averages. The computations in the lifting scheme are done in place, which saves a lot of auxiliary memory and computation time. The lifting scheme can be used to product integer coefficients and so it is exactly reversible. The averages are the coarser representations of the signal and the differences are the extra information required to go from the coarser representation to the original [16]. If the signal is smooth, the detail components will be small and they can be ignored. Then the coarse representation is an approximation of the original and also a compresses version of it with some distortion. The process can be the average component of the signal being the input to the next stage till the number of samples reduces to one. This last sample represents the average component of the entire image. The total number of coefficients after the transform equals the number of coefficients before the transform.

The forward lifting transform shown in the following step:

Step 1: Input signal is first split into even and odd indexed samples. $(Odd_{j-1}, Even_{j-1}) = Split(s_j)$

Step 2: Find a residual difference d_{j-1} between the odd component and its prediction from the even component: $d_{j-1} = Odd_{j-1} - P(Even_{j-1})$ where P is a prediction operator

Step 3: Find a coarse approximation sj-1 of the data by updating the even component $s_{j-1} = Even_{j-1} + U(d_{j-1})$ where U is an update operator.

The inverse transform gets back the original signal by exactly reversing the operations of the forward transform; finally, with a merge in place of a split operation. The number of samples in the input signal must be a power of two and in each succeeding step, it is reduced by half, till the last step produces one sample. The inverse transform shown in the following steps:

Step 1: Reconstruct the even component by reversing the operation in above equation, as follows: $Even_{j-1} = s_{j-1} - U(d_{j-1})$.

Step 2: Reconstruct the odd component by reversing the operation in above equation, as follows: $Odd_{j-1} = d_{j-1} + P(Even_{j-1})$.

Step 3: Combine the odd and even components to generate the data at the previous scale level and repeat the sequence of steps. $s_j = Merge(Even_{j-1}, Odd_{j-1})$.

THE PROPOSED LT-DCT-Q ALGORITHM

The basic steps for implementing the compression process of proposed algorithm are outlined in the following steps:

Step 1: Load the image.

Step 2: Divide the image into non-overlapping M×M blocks.

Step 3: One level of 2d lifting transform is applied to each block of the image.

Step 4: Perform 2D DCT function on low frequency coefficients and discard high frequency coefficients on each block.

Step 5: Quantization of each 8×8 blocks.

Step 6: Perform round of function.

Step 7: Arrange the coefficients using zigzag/adaptive scan.

Step 8: Perform the Huffman coder/Q-coder.

Step 9: Encoding.

Step 10: Compressed image.

Finally, the image is reconstructed followed by inverse procedure. During the inverse LT, zero values are padded in place of the high frequency coefficient. Various cases are used and explained as follow:

Case 1: When using zigzag scan and huffman code.

Case 2: When using zigzag scan and Q-coder.

Case 3: When using adaptive scan and huffman code.

Case 4: When using adaptive scan and Q-coder.

DCT-BASED IMAGE CODING STANDARD: JPEG

The JPEG-DCT is designed for compressing full color or grayscale images of natural, real world scenes. To exploit this method, an image is first partitioned into n×n blocks. A discrete cosine transform (DCT) [17, 18] is applied to each block to convert the gray levels of pixels in the spatial domain into coefficients in the frequency domain. The coefficients are normalized by different scales according to the quantization table. The quantized coefficients are rearranged in a zigzag or adaptive block scan [19] order to be further compressed by an efficient lossless coding strategy such as run length coding, arithmetic coding, or huffman coding. The decoding is simply the inverse process of encoding. So, the JPEG compression takes about the same time for both encoding and decoding. The information loss occurs only in the process of coefficient quantization. The JPEG stand defines a standard 8×8 quantization table [18] for all images which may not be appropriate. To achieve a better decoding quality of various images with the same compression by using the DCT approach, an adaptive quantization table may by used instead of using the standard quantization table.

Adaptive block scanning

For the aim to obtain the best possible compression ratio (CR), discrete cosine tansform (DCT) has been widely used in image and video coding systems, where zigzag scan is usually employed for DCT coefficient organization and it is the last stage of processing a compressed image in a transform coder, before it is fed to final entropy encoding stage. Multiple scanning are used (i.e., vertical, Hilbert, zigzag and horizontal) for different spatial prediction direction on the block. However, due to local prediction errors the traditional zigzag scan is not efficient all time. So we applied a simple and efficient scanning providing [19], for each DCT block vector containing the maximum possible run of zeros at its end. Sorting is the important step of the proposed scan. Descending sort is used for non-zero coefficients. Then the non-zero coefficient is entering to the entropy encoder.

Huffman coding and Q-coder

There are many different reasons for and ways of encoding data and one of these ways is huffman coding [20]. This is used as a compression method in digital imaging and video as well as in other areas. The idea behind huffman coding is simply to use shorter bit patterns for more common characters. A huffman encoder must maintain a data structure that, given a source symbol, returns its code. The decoder must, inversely, return the source symbol corresponding to a code. Q-coder [21] is an adaptive binary arithmetic coder. It has simple adapting mechanism which operates on the estimated probability of coding bin during the coding process. Q-coder avoids the increasing precision problem using fixed precision [21]. Renormalization is applied after interval subdivision to maintain the size and precision of coding interval. Carry propagation during renormalization is solved by ‘bit stuffing’. When 8 successive bits of ‘1’s output, a ‘0’ is stuffed as a carry gap. Multiplication is avoided by using approximated value 1 of rang, because rang is in [0.75, 1.5]. Probability estimation is only taken when renormalization is needed for encoding bin. Computation reduction of interval subdivision and renormalization helps accelerate encoding. However, the precision of arithmetic coding is sacrificed and compression efficiency is degraded. The probability estimation is based on state transition of finite-state machine, which can be implemented as simple table lookup operation.

EPERIMENTAL RESULTS

The LT-DCT-Q image compression algorithm is applied on several images and the results are shown in this section. Several standard grayscale images are used e.g Lena, Barbra and Goldhill images with size 512×512. For lossy compression, distortion was characterized by

a) The peak signal to noise ratio (PSNR), which for an 8-bit decompressed image is defined as [22, 23]:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \qquad \text{(dB)}$$

where MSE refers to the mean squared error between the original image and the reconstructed image.

b) Signal- to- noise ratio (SNR) is defined in the following:

$$SNR = 10 \ \log_{10} \left(\frac{\|X\|^2}{\|X-Y\|^2}\right) \ = 20 \ \log_{10} \left(\frac{\|X\|}{\|X-Y\|}\right)$$

where X an original image and Y is the reconstructed image.

c) Relative mean-square error (RMSE) is computed as

$$\frac{\|X-Y\|^2}{\|X\|^2}$$

The performance of all the above cases can be analyzed using the values obtained from PSNR, CR and SNR which is tabulated below.

Table 1: PSNR in dB and compression ratios (CR) obtained using different bit rate (bpp) for different images (case 1).

	SNR	MSE	PSNR	CR	bpp	Q
Lena512	16.8632	47.3346	31.3790	33.4367	0.2393	25
	18.3713	33.4482	32.8871	21.5950	0.3705	13
	19.3383	26.7717	33.8541	11.3495	0.7049	5
Barbara	10.3448	205.7756	24.9969	26.0713	0.3069	25
	10.7362	188.0412	25.3883	17.2948	0.4626	13
	10.9311	179.7901	25.5831	10.0433	0.7966	5
Goldhill	14.9961	76.6987	29.2829	27.8414	0.2873	25
	16.3500	56.1569	30.6368	16.5048	0.4847	13
	17.1711	46.4830	31.4579	9.6838	0.8261	5

Table 2: PSNR in dB and compression ratios (CR) obtained using different bit rate (bpp) for different images (case 2).

	SNR	MSE	PSNR	CR	bpp	Q
Lena512	16.8693	47.2681	31.3851	63.7393	0.1255	25
	18.3860	33.3352	32.9018	40.4395	0.1978	13
	19.3631	26.6187	33.8789	37.2212	0.2149	5
Barbara	10.3645	204.8458	25.0165	72.1365	0.1109	25
	10.7596	187.0331	25.4116	53.1194	0.1506	13
	10.9565	178.7427	25.6085	32.1225	0.2490	5
Goldhill	14.9972	76.6795	29.2840	55.2362	0.1448	25
	16.3555	56.0852	30.6423	48.5295	0.1648	13
	17.1836	46.3496	31.4703	35.3639	0.2262	5

Table 3: PSNR in dB and compression ratios (CR) obtained using different bit rate (bpp) for different images (case 3).

	SNR	MSE	PSNR	CR	bpp	Q
Lena512	19.3529	26.6815	33.8687	16.5888	0.4823	5
	18.3479	33.6290	32.8637	32.8872	0.2433	13
	16.8417	47.5700	31.3575	54.3191	0.1473	25
Barbara	10.9545	178.8253	25.6065	13.7400	0.5822	5
	10.7473	187.5647	25.3993	26.8835	0.2976	13
	10.3583	205.1394	25.0103	46.1288	0.1734	25
Goldhill	17.1513	46.6948	31.4381	13.1514	0.6083	5
	16.3394	56.2938	30.6262	27.3508	0.2925	13
	14.9861	76.8759	29.2729	50.2601	0.1592	25

The results are also compared with the standalone JPEG, JPEG2000 and SPIHT algorithms, as shown in figure 2 and figure 3. The results indicate that, in the case of standard images one can achieve high PSNR values and better image quality for both SPIHT and LT-DCT-Q scheme compression technique.

Table 4: PSNR in dB and compression ratios (CR) obtained using different bit rate (bpp) for different images (case 4).

	SNR	MSE	PSNR	CR	bpp	Q
Lena512	18.9244	29.4483	33.4402	37.5114	0.2133	1.5
	19.5454	25.5246	34.0612	49.1505	0.1628	3
	19.3529	26.6815	33.8687	68.1026	0.1175	5
	18.9754	29.1044	33.4912	88.8548	0.0900	8
	18.3479	33.6290	32.8637	105.464	0.0759	13
Barbara	10.9545	178.8253	25.6065	59.1280	0.1353	5
	10.8896	181.5188	25.5416	69.3778	0.1153	8
	10.7473	187.5647	25.3993	95.7604	0.0835	13
Goldhill	17.1513	46.6948	31.4381	57.9356	0.1381	5
	16.8878	49.6155	31.1746	77.8858	0.1027	8
	16.3394	56.2938	30.6262	99.6035	0.0803	13

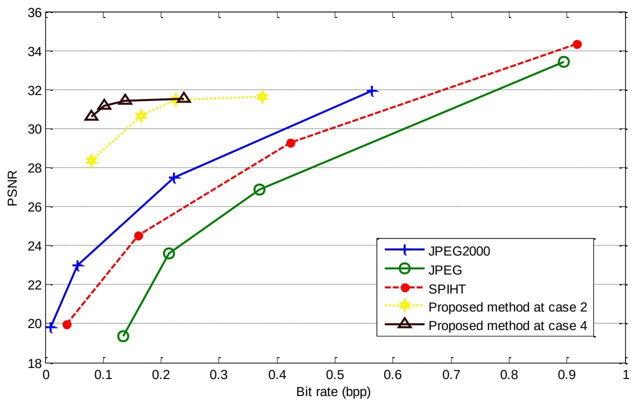

Fig. 2 Comparison results of various standard methods for Goldhill image

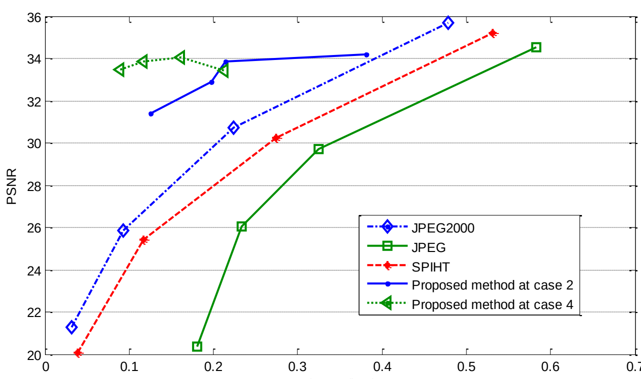

Fig. 3 Comparison results of various standard methods for Lena image

The analysis of the chart shows that for a fixed level of distortion, the number of bits required to transmit the hybrid coefficient would be less than those required for other algorithm.

The original varies images (figure 4 (a)) and the reconstructed images using the two different compression schemes manely SPHIT algorithm and proposed scheme is shown in figure 4. it is clearly evident from the figure that reconstructed different images show better quality of the image for proposed scheme (figure 4) compared to that of SPIHT scheme (figure 4 (d)) as the PSNR values achievable for proposed scheme is much higher than that of SPIHT scheme. It is evident from the figure 4(b, c) that the proposed scheme is must efficient technique for compression of different images. PSNR values achieved for these images are very high and the quality of the reconstruction of the images is much better than those obtained using SPIHT algorithm. Further the results of present scheme illustrate that for different images high PSNR values are achieved for LT-DCT-Q scheme at much lower bitrates.

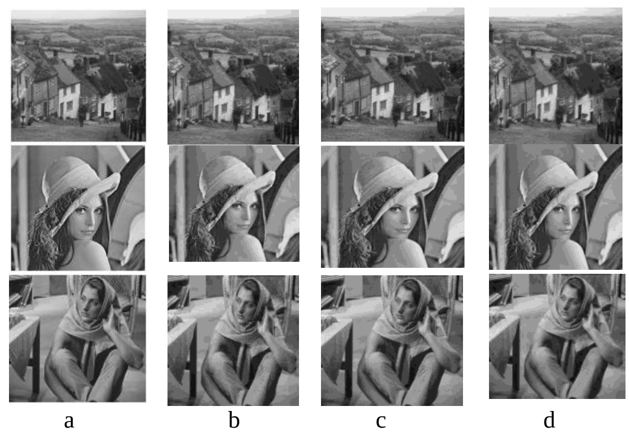

Fig. 4: a) The original image for Goldhill, Lena and Barbara, b) Decoded images by using case 2 at Q=25, c) Decoded images by using case 4 at Q=8, d) Decoded images SPIHT at 0.25 bpp.

CONCLUSION

In this paper a novel image compression technique for efficient storage and delivery of data is proposed. It is based on decomposing the data using lifting scheme in combination with the discrete cosine transform and the Q-coder. Experimental results show that the LT-DCT-Q scheme outperforms JPEG based DCT, JPEG-2000 and the SPIHT coding algorithm. For a very low bit rate, for example 0.125 bpp or lower, the proposed LT-DCT-Q approach is superior to other approaches. The proposed method (case 4) provides higher compression ratio and similar peak signal to noise ratio compared to the other lossy image compression schemes. Higher compression ratio infers to higher compression performance.

REFERENCE

1. Yan Wang, Shoushun Chen, Bermak, A., "FPGA implementation of image compression using DPCM and FBAR," *International Conference on Integrated circuits, (ISIC'07)*, 329-332 (2007).
2. Xiwen Owen Zhao, Zhihai Heny He, "lossless image compression using super-spatial structure prediction," *IEEE Signal Processing Letters*, **17**(4): (April 2010).
3. Baluram N., Farukh H., Pradeep D., "Comparative analysis of fast wavelet transform for image compression for optimal image quality and higher compression ratio,", *International Journal of Engineering Science and Technology (IJEST)*, **3**(5) (May 2011).
4. David S., Data compression," *Springer-Verlag New York 3rd Edition* (2005).

5. Aaron T. Deever, Sheila S. Hemami, “Lossless image compression with projection-based and adaptive reversible integer wavelet transform,” *IEEE Transactions on Image Proceesing*, **12**(5) (May 2003).
6. Nikolaos V. Boulgouris, Dimitrios Tzovaras, Michael Gerassimos Strintzis, “Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding,” *IEEE Transactions on Image Processing*, **10**(1) (Jan. 2001).
7. Eddie Batista de Lima Filho, Eduardo A. B. da Silva Murilo Bresciani de Carvalho, Frederico Silva Pinage, Universal “Image compression using multiscale recurrent patterns with adaptive probability Model,” *IEEE Transactions on Image Processing,* **17**(4) (Apr. 2008).
8. [8] Sachin D., “A Review of image compression and comparison of its algorithm,” *International Journal of Electronics & Communication Technology (IJECT)*, **2**(1): 22-26 (Mar. 2011).
9. [9] Janaki. R. Dr.Tamilarasi, “A Still image compression by combining EZW encoding with huffman encoder,” *International of Computer Applications (0975 – 8887)*, 13(7) (Jan. 2011).
10. [10] H. Singh, S. Sharma, “Hybrid image compression using DWT, DCT & Huffman encodind technique,” *International Journal of Emerging Technology and Advanced Engineering*, **2**(10): 300-306 (Oct. 2012).
11. [11] Sriram B., Thiyagarajan S., “Hybrid transformation technique for image compression,” *Journal of Theoretical and Applied Information Technology*, **41**(2): 175-180 (July 2012).
12. [12] Hao Xu, Jizheng Xu, Feng Wu, “Lifting-based directional DCT-like transform for image coding,” *IEEE transaction on Circuits and system for Video Tecnology*, **17**(10): 1325-1334 (Oct. 2007).
13. [13] Wenbing Fan, Jing Chen, Jina Zhen, “SPHIT algorithm based on fast lifting wavelet transform in image compression,” *Computational Intelligence and Security Lecture Notes in Computer Science*, 2802: 838-844 (2005).
14. [14] Swelden, “A Fast progressive image sampling using lifting scheme and non-uniform B-splines,” *IEEE International Symposium on Industrial Electronics (ISIE 2007)*: 4-7 (June 2007).
15. [15] R. ClayPoole, G. Davis, W. Sweldens, R. Baraniuk, “Nonlinear wavelet transform for image coding via lifting,” *IEEE transaction on image processing*, (1999).
16. [16] G. Davis, A. Nosratinia, “Wavelet-based image coding: An overview,” *Applied and Computational Control, Signala and Circuits*, **1**(1) Spring (1998).
17. [17] Rao K. R., Yip P., Discrete cosine transform- Algorithms, Advantages, Applications,” Academic Press (1990).
18. [18] Wallace G. K., “The JPEG still picture compression standard,” *Comm. ACM*, **34**(4):30-40 (April 1991).
19. Walaa M. Abd-Elhafiez, Usama S. Mohammed, A. Akilicman,” On a high performance image compression technique,” *Journal of ScienceAsia, 39(4): 416-422, doi: 10.2306/scienceasia 1513-1874.2013.39.416* (2013).
20. D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *In Proccing I.R.E.*, **40**: 1098-1101 (1952).
21. W. B. Pennebaker, J.L.Mitchel, G. G. Langdon, Jr. R. B. Arps, “An overview of the basic principle of the Q-coder adaptive binary arithmetic coder,” IBM Journal of research and Development, 32(6): 717 (1988).
22. M. Rabbani, R. Joshi, “An overview of the JPEG200 still image compression standard,” Signal Processing: Image Communication, 17: 3-48 (2002).
23. Walaa M. Abd-Elhafiez, Wajeb Gharibi, “Color Image Compression Algorithm Based on the DCT Blocks”, International Journal of Computer Science Issues (IJCSI), Volume **9**, Issue 4, May (2012).