ISSN 2278-3083
Volume 8, No.6, November –December 2019
International Journal of Science and Applied Information Technology
Available Online at http://www.warse.org/ijsait/static/pdf/file/ijsait15862019.pdf
https://doi.org/10.30534/ijsait/2019/158620198

A HYBRID IMAGE COMRESSION TECHNIQUE USING DFT AND DCT

Aesha A. Amhammed, Mahmoud Omari
Department of Computer Science, Amman Arab University, Amman, Jordan

ABSTRACT

The main aim of image compression is to decrease the size of the data required to represent an image. Thus, reducing the required space to store the image and reducing the time required to transfer the image throughout computer networks. The art of designing and developing an image compression system is a compromise between image quality and compression ratio. This research proposes a new hybrid image technique to achieve high compression ratio and at the same time saving the quality of the compressed image. The proposed technique combines the benefits of two transformations; the Discrete Fourier Transform (DCT) and the Discrete Cosine Transform (DCT). The experimental results show that the suggested approach preserves the quality of the reconstructed image and achieved higher compression ratio compared to individual standalone of the aforementioned transformations. The results of the proposed hybrid technique show that the average value of PSNR over all test images is 29.89 compared to those of DFT and DCT which are 33.07 and 32.13, respectively. The results also show that the hybrid technique achieved 23.81 average compression ratio over all test images compared to 19.93 and 21.75 for DFT and DCT, respectively. It is clear that the proposed technique outperforms both standalone DFT and DCT in terms of compression ratio and at the same time preserves the quality of the compressed image.

Key words: Image processing, compression, hybrid, DFT, DCT

1. INTRODUCTION

With the rapid development of technology and communication, and with the extensive use of image transmission over the Internet, it became necessary to find an effective method to reduce storage space requirements and accelerate the transfer of images. Image compression is necessary and very important to help solve these problems. Compared to text data, Images requires high storage space and long transmission times. Image compression exploits the redundancy in data to reduce the storage space needed to store the image. Reducing repetition in image details is the primary goal of image compression (Song et. al., 2008).

Image compression can be divided into two types: lossy and lossless compression. Lossy compression achieves a high compression ratio but loses some details of the image after reconstructing it. In the other hand, lossless compression achieves lower compression ratio compared to lossy compression but maintains the details of the image after being reconstructed. Most image compression techniques are based on some type of transformation, which constitutes the main component of modern image and video processing applications. The transformation coding depends on the evidence that pixels in an image demonstrate some level of correlation with their adjacent ones. Thus, these correlations con be utilized to make a prediction of the value of a pixel from its neighboring ones. A transformation can be defined as a mapping function that maps correlated spatial data into uncorrelated transformed coefficients. Clearly, the transformation must exploit the fact that the contribution of individual pixel to the whole image is relatively small. That is, the visual contribution of individual pixel can be predicted throughout its neighbors.

A typical image compression and transmission system is shown in Figure 1. The purpose of the source encoder is to make use of the redundancies in image data to offer compression. In other words, the source encoder decreases the average number of bits needed to represent the image, which means a reduction in the image entropy.

Some of the most commonly used lossy compression algorithms are the Discrete Cosine Transforms and Discrete Fourier Transforms. The DCT algorithm has high energy compaction property and requires less computational resources. The energy compaction property of an algorithm refers to the ability to concentrate most important information signal into as much as few low frequency components. On the other hand, DFT is a Highly efficient algorithm for estimating Discrete Fourier Transforms that have been developed since the mid60's. These transforms are known as Fast Fourier Transform algorithms and they rely on the fact that the standard involves a lot of redundant calculation. The main disadvantages of are the introduction of false contouring effects and blocking artifacts at higher compression, while DFT maintains the quality of the compressed image to a large extent. Therefore, the idea of exploring the advantages of both algorithms motivated us to investigate combination of DFT and DCT algorithms. Such combination of two algorithms is referred to as "Hybrid" scheme. Given the need to reduce the storage space and reduce the transmission time needed, image compression is the best solution to address these problems.

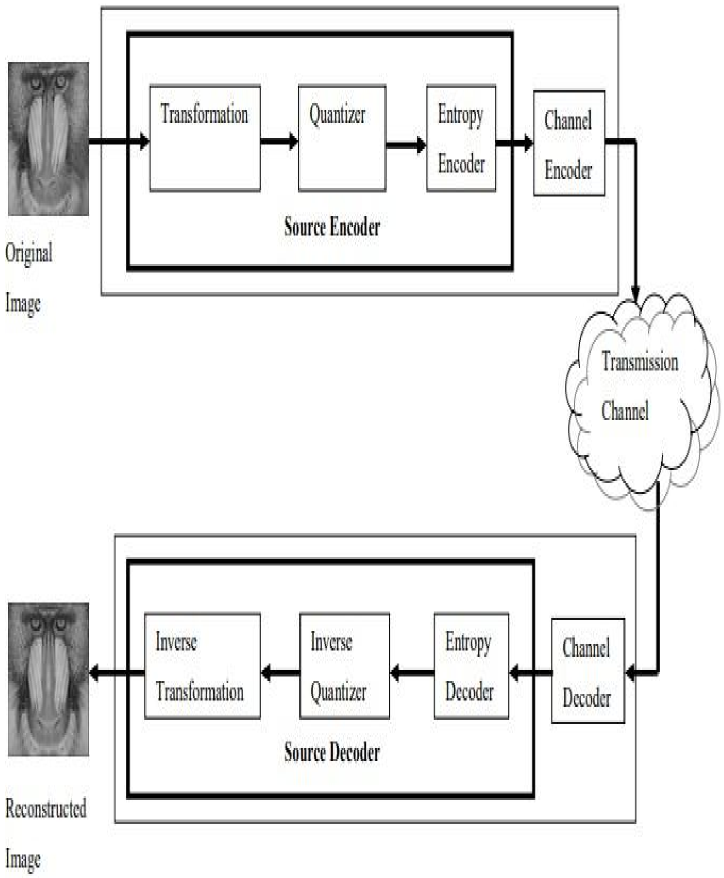

Figure 1: Components of a Typical Image Transmission System (Radha, 2003).

This paper is organized as follows. The following two sections present an overview of both DFT and DCT transformations as image compression methods. The proposed method is presented in section 4. Experimental results an analysis are presented in section 5. Section 6 concludes the paper.

2. DISCRETE FOURIER TRANSFORMATION

The fast Fourier transform (FFT) is considered a sub-optimal algorithm to compute the DCT. The computational complexity for the DFT of size N requires O(N2) operations, while the FFT needs O(NlogN) operations to achieve the same task. The FFT exploits the algebraic redundancies inherent in the DFT.

The DFT transforms a set of finite data elements in one or more dimensions into another finite set of data. In general, DFT transforms a sequence of N data samples {fn: $0 \leq n \leq$ N-1} into another sequence {Fk: $0 \leq k \leq$ N-1} of numbers that represent the Fourier coefficients of the original data samples. The transformation is reversible and allows an alternative view of information that might be very useful in some applications such as compression of images and sound signals. In image compression, DFT transforms image information from spatial domain to frequency domain. The Fourier transform is used in a wide range of applications, such as image filtering, image analysis, image reconstruction and image compression. Fourier transforms are very useful at providing frequency information that cannot be seen easily in the time domain. However, they do not suit brief signals that change suddenly or in fact any nonstationary signals. The reason is that they show only what frequencies occur, not when these frequencies occur, so they are not much of a help when both time and frequency information are required simultaneously. In stationary signals, all frequency components occur at all times, so Fourier transform is very useful. The frequency domain representation represents exactly the same signal, but in a different form. Using the Inverse Fourier transformation, the converted signal can be restored from frequency dimension [14]. To compute all N Fourier coefficients using DFT, the following formula is used [5]:

Where i is the complex number defined by i2= -1. As mentioned before, the computation of N Fourier coefficients using the above formula will cost a total of O(N2). Alternatively, FFT can be used, which is an efficient algorithm to calculate the DFT with low computational complexity. While FFT produces the same result as DFT, it is incredibly more efficient, often reducing the computation time to O(NlogN). The following formula is used to compute the inverse DFT [5]:

In image compression applications, for example, a gray scale image of M by N pixels is represented by a 2-dimessional array of pixel values f(m ,n), where m = 0 .. M-1 and n = 0 … N-1. In this case, two dimensional DFT is needed to compute the Fourier coefficients for the image. To do that, first, computing a 1-dimenssional FFT of length N of the rows of the input image array then by applying a 1-dimessional FFT of length M on the columns.

3. DISCRETE COSINE TRANSFORMATION

Discrete cosine transform transforms a signal into fundamental frequency components [10]. One advantage of using the DCT is that the results are real numbers whereas the DCT results are complex numbers. This transformation is widely used in many applications in engineering and science. Discrete cosine transform is vastly used for video and image compression and is comparable to the discrete Fourier transform. The Discrete cosine transform is computationally easier than discrete Fourier transform beside having best energy compaction properties than the discrete Fourier transform. This means that more information is encoded by the discrete cosine transform coefficients than by the discrete Fourier transform coefficients. This may initially appear inconsistent, that discrete cosine transform ignores the odd sine components of a function and yet appears to encode more information. For this reason, the discrete cosine transform is used in video and image compression systems such as JPEG[11]. JPEG stands for Joint Photographic Experts Group, which was a group of expert parties in image processing that devised a standard for image compression [8]. The JPEG algorithm design was specific for the human eye which is more sensitive to the illuminance part of the image, rather than the chromatic value. In addition, the eye is not sensitive to high-frequency content in image [19]. In a typical image, the DCT causes most visually significant information about the image to be concentrated using two coefficients of the DCT. Within the frequency domain, the DCT coefficient that is located in the upper-left of image block signifies the image's low-frequency information and its rough contour. At the same time, the DCT coefficient that is located in the lower right of the image signifies the image's high frequency information and its fine texture. The high frequency coefficients can be discarded on the basis of tolerable resolution loss. The general equation for a 2D (N by M image) DCT is defined by the following equation [5]:

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i).A(j).\cos\left[\frac{\pi.u}{2.N}(2i+1)\right]\cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$

Where,

$$A(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0 \\ 1 & otherwise \end{cases}$$

The basic operations of compressing an image using DCT are as follows: The input image is divided into 8×8 blocks of pixels. For each block, the DCT coefficients are computed then these coefficients quantized using a predefined 8x8 quantization matrix. Quantization is done by dividing each coefficient by a specific constant followed by rounding to nearest integer. This process results in many zeroes for those high-frequency coefficients which allows to reduce the amount of information needed to represent the image. The resulted 8x8 matrix is then encoded using some entropy encodes such as Huffman Code [6] to get the resulted compressed image. To decompress the image, for each bock a decoding step is performed to reverse the Huffman encoding followed by de-quantization to get the DCT coefficients and computing the two-dimension inverse discrete cosine transform. During the compression process of a typical image, many of the DCT coefficients become zero after quantization; thus, a great part of the image representation having repeated zero values.

4. THE PROPOSED HYBRID TECHNIQUE

This research is intended to evaluate the compression performance of a proposed hybrid compression technique that combines both DFT and DCT image compression algorithms. There have been many attempts to exploit different compression methods by constructing a hybrid method using different transformations and approaches [1-4] [9] [13]. Each attempt tries to increase the compression ratio and at the same time wishing to preserve the quality of the image. The general architecture of the suggested approach is outlined in Figure 2.

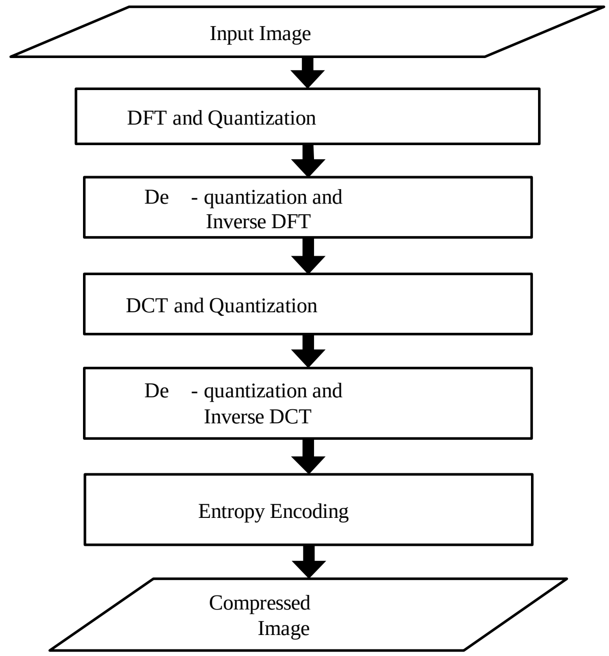

Figure 2: The Proposed Hybrid Image Compression Method.

As shown in Figure 2, the proposed scheme starts by applying the DFT algorithm in the first stage, discarding the high frequency coefficient through quantization and de-quantization, and then performing the inverse DFT to construct a compressed image. Similarly, the second stage is applying the DCT on the resulted image to compress it once more.

The input image in the two stages consists of three planes representing the Red, Green, and Blue color components of the image. Each plane is a two-dimensional matrix of pixel values ranging from 0 to 255. The following steps is performed on each plane of the image. The first step in the first stage is splitting the image planes into 8×8 pixel blocks. For each block, the DFT coefficients are computed by the Fourier transform function. A quantization process is applied on the coefficients to keep the coefficients that constitute the first four rows and the first four columns of the resulted matrix and zeroing the remaining coefficients. After getting rid of unimportant coefficients, the remaining ones are considered for the reconstruction of the image.

To produce the reconstructed image and make it ready for the second stage, an inverse DFT is performed to generate the image, which is again represented in three planes for the RGB color components of the image. The resulted image from the first stage is now ready as an input for the second stage, the DCT stage. Similarly, in the DCT stage, the input image is divided into 8x8 pixel blocks. The DCT coefficients are computed by applying the DCT function that produces an 8x8 coefficients matrix with high and low frequencies. The coefficients are quantized by dividing the coefficients matrix by a predefined quantization matrix used in JPEG compression technique, by which, the high frequency coefficients are given low priority in representing the image and zeroed. To store the compressed image in a file or prepare it for network transmission, a de-quantization process is performed to restore the original coefficients matrix. The restoration of the original coefficients is done by multiplying the resulted matrix by the quantization matrix. An inverse DCT is performed on the restored coefficients to restore an approximation of the original pixel values. The restored pixels' values are then reorganized in consistent with zig-zag arrangement, keeping the zero values at the end of the list. Finally, an entropy encoder, particularly Huffman code, is used to store the image.

5. EXPERIMENTAL RESULTS AN ANALYSIS

The experiments were carried on several images commonly used in the literature to compare the different image compression techniques. These images have different characteristics suitable for stimulating the different compression techniques. It is well known in the research community specialized in image compression that some images among other ones are the most commonly used to test image compression algorithms. These standard test images are: Baboon, pepper, Barbara, fruits, Tulips, girl, airplane, Gold Hill, Lena, and Zelda. All these images are color 24-bit per pixel in uncompressed bitmap (bmp) format and of various sizes (256x256, 512x512, …, etc.). To examine the effectiveness and efficiency of the proposed scheme, the following measures are used: Compression ratio, Peak Signal to Noise Ratio (PSNR), and Mean Square Error (MSE).

The compression techniques, DFT and DCT algorithms and the proposed Hybrid technique are implemented using MATLAB R2015a image processing software under 64-bit MS Windows environment. The performance of all these compression methods is measured by applying them on the same standard test images. By using the same standard test images, the result of this research can be compared both visually and quantitatively. Those standard test images are downloaded from one of the databases available at [7]. MATLAB code has been written for each experiment and the results of the evaluation metrics are collected for each image. The following is a summarization of the steps of the proposed hybrid technique:

1) Read the original 3-component (i.e., RGB) uncompressed image from a bmp format file and store it in a corresponding matrix.
2) Crop the image so that the size of each of the two dimensions of the image are multiples of 8.
3) Apply the DFT process which consists of the following steps:
a) Divide the image matrix into 8x8 blocks.
b) Compute the DFT coefficients for each block.
c) Quantize the coefficients by keeping only those coefficients that lie in the first four rows and the first four columns.
d) Perform inverse DFT to reconstruct an approximation of the original image after compression.
4) Apply the DCT process, on the resulted image from the previous step, by performing the following operations: a) Subdivide the image matrix into 8x8 blocks.
b) Compute the DCT coefficients for every block.
c) Quantize the DCT coefficients based on quantization matrix to discard high frequency values: The block matrix is divided by the quantization matrix and round the results.
d) Perform de-quantization process to restore the DCT coefficients: multiply the result by the quantization matrix.
e) Perform inverse DCT to restore approximation of the original pixels' values.
5) Compress the image data by Huffman code entropy encoder to store the resulted compressed image.

The performance analysis is performed extensively for the DFT and DCT algorithms and the proposed hybrid technique. The analysis is done objectively using the standard metrics, MSE, PSNR, and compression ratio. A higher value of PSNR is better because it means that the ratio of signal (i.e., the original image) to noise (i.e., error in the reconstruction) is higher. MSE is the cumulated squared error between the original the compressed images. Therefore, a lower value for MSE means less error.

Table I shows the PSNR and MSE values for all the standard test images using DFT, DCT, and the proposed hybrid technique. The first thing that can be noticed from the results in the table is that all the aforementioned compression schemes achieved low PSNR values for the Baboon image. Baboon image is one of the important standard test images that has distinctive mix of colors and textures which represents a challenge for any compression algorithm. When visually inspecting the reconstructed image of the Baboon, a person cannot distinguish any differences. This is due to the fact that there are no big differences between the PSNR values for DFT, DCT, and the hybrid techniques, which are 28.38, 25.84, and 24.35, respectively.

Table1: PSNR AND MSE VALUES OF THE THREE COMPRESSION TECHNIQUES

	DFT		**DCT**		**Hybrid**	
Image Name	***PSNR***	***MSE***	***PSNR***	***MSE***	***PSNR***	***MSE***
Baboon512	28.34	95.39	25.84	169.33	24.35	238.59
Pepper512	33.36	30.00	31.89	42.06	30.03	64.61
Barbara720x576	31.77	43.31	31.53	45.76	28.91	83.53
Fruits512	33.66	28.01	33.52	28.91	30.87	53.25
Tulips768x512	31.78	43.21	33.69	27.80	29.92	66.30
Girl512	35.53	18.19	32.95	32.98	31.45	46.54

Airplane512	32.54	36.23	34.07	25.48	30.63	56.25
Golghill720x576	34.25	24.44	32.22	39.03	30.50	57.96
Lena512	35.63	17.80	33.05	32.20	31.56	45.36
Zelda256	33.85	26.81	32.55	36.15	30.65	55.97
Average	**33.07**	**36.34**	**32.13**	**47.97**	**29.89**	**76.84**

Once again, by inspecting the PSNR results in Table 1, one can notice that the highest PSNR value of 35.63 achieved by DFT is for Lena image and for the same image the hybrid technique also achieved the highest PSNR value of 31.56, while the DCT achieved a PSNR value of 33.05. The Lena image has been used widely as an image for testing compression algorithms due to its distinctive features: the hat contains repetitive parallel weave structures; the feather contains complex textures; the skin of the face and shoulder show smooth intensity gradations; and the eyes include familiar small-scale features in the iris and lashes. With these complex features in the Lena image, all those compression schemes performed comparable PSNR results. Although there are small differences in the PSNR values for all the compression techniques under consideration, by looking into the reconstructed Lena images for all those compression schemes, a person cannot notice any visual differences. Considering all the standard test images, the average PSNR value achieved by all the compression schemes are as follows: DFT averaged 33, the DCT averaged 32, and the hybrid technique averaged 30. The PSNR and MSE values obtained show that the proposed method has similar image quality to those of DFT and DCT. The proposed method takes the advantages of both DFT and DCT algorithms while preserving the quality of the reconstructed image.

Table II shows the compression ratio of the three techniques under consideration. It can be noticed that the hybrid technique outperforms both DFT and DCT in compression ratio for all the standard test images. Looking at the results in Table II, the lowest compression ratio for all compression schemes resulted for the Baboon image. As said before, the Baboon image has complex structure and features that represent a challenge for the compression algorithms, resulted in a low compression ratio for all techniques. In the other hand, the compression techniques achieved the highest compression ratio on the Girl image, which is a simple image that does not have complex features.

Table 2:. COMPRESSION RATIO OF THE THREE TECHNIQUES

Image Name	**DFT**	**DCT**	**Hybrid**
Baboon512	12.26	12.69	14.36
Pepper512	20.83	23.25	25.46
Barbara720x576	17.44	18.36	19.81
Fruits512	22.29	24.49	27.35
Tulips768x512	16.93	17.58	19.29
Girl512	23.34	26.34	28.88
Airplane512	21.91	24.04	26.08
Golghill720x576	20.01	22.14	24.11
Lena512	22.79	25.25	27.56
Zelda256	21.47	23.32	25.23
Average	**19.93**	**21.75**	**23.81**

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this research, a new hybrid image compression technique is proposed in order to have an effective compression. Initially, the discrete Fourier transform is applied to compress the image, then, the discrete cosine transform is applied to compress the image once more. In both algorithms, the image is segmented into (8x8) blocks and the DFT and the DCT coefficients are calculated for every single block. Next, quantization was performed according to the quantization table. After that, the quantized values were then reorganized in consistent with a zigzag arrangement. The lower values were rejected from the list in this arrangement. After getting rid of unimportant coefficients, the remaining ones were compressed by entropy encoder.

In order to verify the efficiency of the proposed algorithm, several experiments were carried out and the results were compared with those of standalone DFT and DCT. The PSNR values of the proposed technique are comparable with those of DFT and DCT. The experimental results reveals that the proposed hybrid technique outperforms both the DFT and the DCT with respect to compression ratio.

This research brings to light some possible opportunities for further research; such as using different quantization methods to obtain higher compression and at the same time hoping to preserving the image quality.

REFERENCES

1. Bansal, N., & Dubey, S. (2013). Image compression using hybrids transform technique. International Journal of Global Research in Computer Science, 4(1), 13-17.
2. Chang, C., & Tai, W. (2007). Reversible hiding in DCT-based compressed images. Information Sciences, 177(13), 2768-2786

3. Chung, K., & Yan, W. (2006). A hybrid gray image representation using spatial-and DCT-based approach with application to moment computation. Journal of Visual Communication and Image Representation, 17(6), 1209-1226.
4. Elharar, E., & Javidi, B. (2007). A hybrid compression method for integral images using discrete wavelet transform and discrete cosine transform. Journal of display technology, 3(3), 321-325. https://doi.org/10.1109/JDT.2007.900915
5. Geddes, K. (2008, August), "Course Notes for Introduction to Scientific Computation," University of Waterloo.
6. Huffman, D. (1952). "A Method for the Construction of Minimum Redundancy Codes" (PDF). Proceedings of the IRE. vol. 40, no. 9, pp. 1098–1101.
7. Image Databases. (2018, January). Retrieved from http://www.imageprocessingplace.com/root_files_V3/image_databases. htm/
8. JPEG. "Joint Photographic Experts Group, JPEG Homepage". Retrieved 2019-03-19.
9. Krikor, L., & Shaaban, Z. (2009). Image encryption using DCT and stream cipher. European Journal of Scientific Research, 32(1), 47-57.
10. Nageswara, R., & Srinivasa, K. (2008). Image compression using discrete cosine transform. Computer Sciences and Telecommunications, (3), 35-44.
11. Ra, R. K. and P. Yip, “Discrete Cosine Transform: Algorithms, Advantages and Applications.”, NY: Academic, 1990.
12. Radha, H. (2003, January), “Lecture Notes: ECE 802 - Information Theory and Coding”
13. Abualigah, L. M. Q. (2019). *Feature selection and enhanced krill herd algorithm for text document clustering*. Berlin: Springer.Russ, J. (2011). Processing binary images. The image processing handbook, 4, 410-412. https://doi.org/10.1007/978-3-030-10674-4
14. Saupe, D. (1996). A new view of fractal image compression as convolution transform coding. IEEE Signal Processing Letters, 3(7), 193195.
15. Singh, H., & Sharma, S. (2012). Hybrid image compression using DWT, DCT and Huffman encoding techniques. International Journal of Emerging Technology and Advanced Engineering, 2(10), 300-306.
16. Truong, T., & Li, A. Q. (2000). A fast encoding algorithm for fractal image compression using the DCT inner product. IEEE transactions on Image Processing, 9(4), 529-535. https://doi.org/10.1109/83.841930
17. Veerpal, G. (2012). Hybrid Image Compression Using DWT, DCT and Huffman Coding Techniques. International Journal of Advanced Research in Computer Science and Electronics Engineering, 2(10), 22502459.
18. William B. Pennebaker; Joan L. Mitchell (1993). “JPEG still image data compression standard”, (3rd Ed.). Springer. p. 291. ISBN 978-0-44201272-4.