JPEG Image Compression Applying 2D Discrete Cosine Transform

Sumit Poonia[1], Vijaypal Singh[2]

[1]*M.Tech Scholar, Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India*

[2]*Assistant Professor, Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India*

Abstract: This paper proposed the work that how image compression is achieved using two dimensional discrete cosine transform. As we can see in today's scenario of world use of images is expanding day by day in various applications. And we have limited storage capacity and bandwidth for transmitting the image data, so to save these two we apply the concept of image compression. As compressed image requires less storage space for storing and less channel bandwidth for transmission. There are various methods of image compression but here we will use DCT which is mostly use for JPEG images. As PSNR is a quality check parameter and in this paper we will check PSNR of images which we get after different block sizes used in DCT for compressing the image.

Keywords: *JPEG, DCT, image compression, PSNR, MSE.*

I. INTRODUCTION

In image compression data required for storage and transmission of image is reduced up to much extent. Image compression can categorised mainly into two categories [2][5] i.e. either it will be lossy compression or lossless compression. If we talk about lossy compression the size of image is reduced with the compensation of some details of image i.e. some image details are got destroyed that can never be extracted or decoded from the image that has compressed [8] whereas in other case that is in lossless compression all the necessary details are preserved along with compression [5]. Lossless compression is used where all the details of image are very important like in medical imaging where we cannot compromise with image details for compression [3][5]. But in our daily life mostly we use lossy type of image compression. In lossy compression some details of image are also distorted but the compression ratio becomes very high [1][2]. In this paper we will perform compression on JPEG format images.JPEG stands for Joint Photographic Group of Experts and also called as JPG [8]. This image format uses lossy type of image compression. In this firstly 2D discrete cosine transform is performed on the input image [6][7] then further operations are performed on the blocks with size 8x8 or 16x16 or 32x32.

A. Image Compression

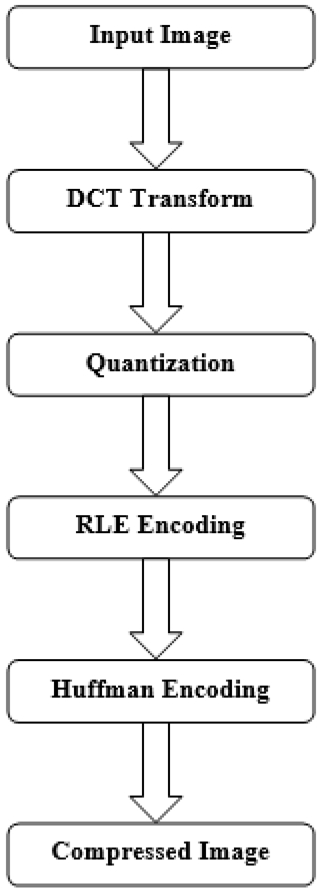

Fig 1: Image Compression flow chart

DCT: A Discrete Cosine Transform is a representation of finite sequence of different data points as total sum of cosine function only with oscillations at variable frequencies**.** DCT is used in image compression as it removes very small and high frequencies from the image. Equation for 1D-DCT is:

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{M-1}) + \sum_{m=1}^{M-2} x_m \cos\left[\frac{\pi}{M-1} mk\right]$$

$$k = 0, \dots, M-1 \quad (1)$$

Equation for 2D-DCT is:

$$X_k = \sum_{m=0}^{M-1} x_m \cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k\right]$$

$$k=0,\ldots,M-1 \qquad (2)$$

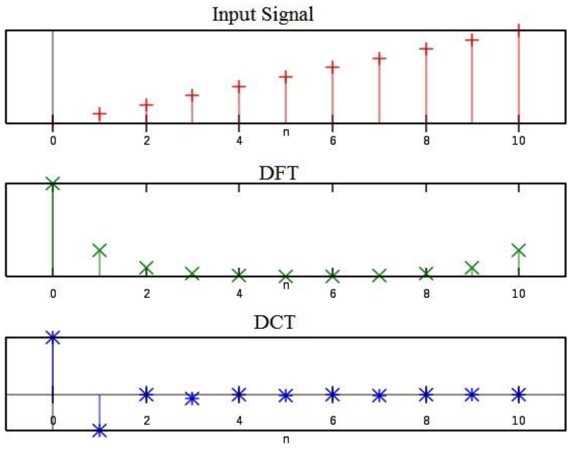

Fig 2: Discrete Cosine Transform

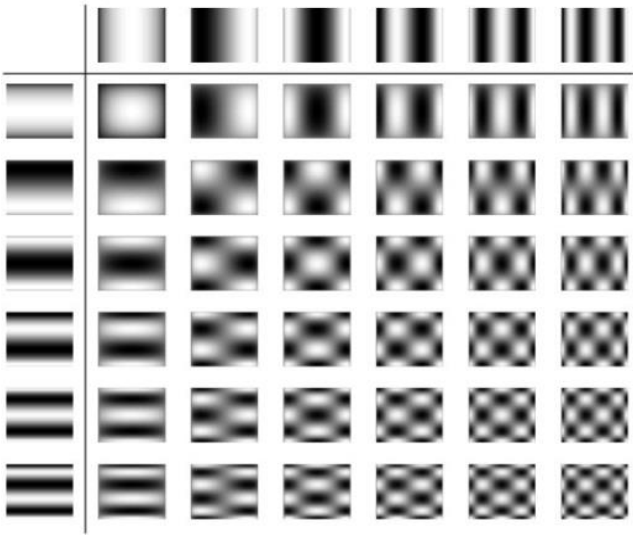

Fig 3: 2D DCT Frequencies from DCT

Quantization: It is a lossy type image compression technique in which large no. values are compressed to desired limited number. It can be of two types i.e. colour quantization and frequency quantization. In colour quantization large no. of colours are reduced to limited no. of colours while in frequency quantization high value frequencies are compressed as human eye can differentiate even small brightness variations at small frequencies but unable to detect high frequency (varying rapidly) brightness variations.

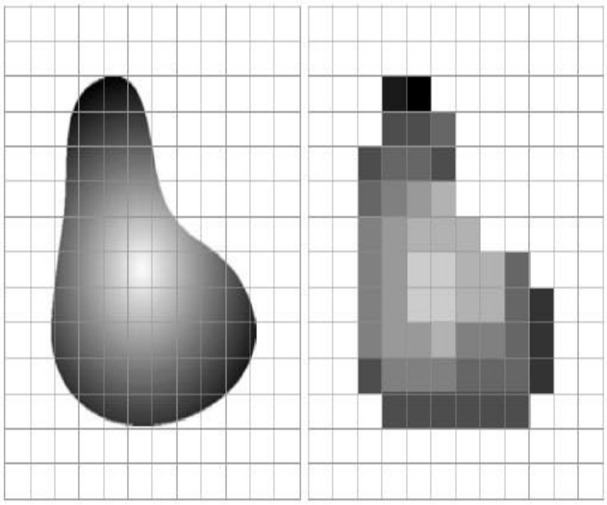

Fig 4: Quantization of colours

RLE: It stands for Run Length Encoding, it is a general form of data compression (using lossless type) where runs of data (i.e. sequences which contains similar data values which occur with consecutive data values) are stored in the form of single data value, instead of the original run. For example:

RRRRRRRRLLLLEEEEELLLRRRRRREEE

Above line can be renders using RLE as:

8R4L5E3L6R3E.

Huffman Encoding: Huffman coding is a particlar type of optimized prefixed code that is used for lossless type compression of data. This algorithm is designed by David A. Huffman who was studying at MIT and published it in paper "A Method for the Construction of Minimum-Redundancy Codes" in 1952. The algorithm gives a variable length code at output by estimating probability or frequency of occurring every possible values from source symbol.

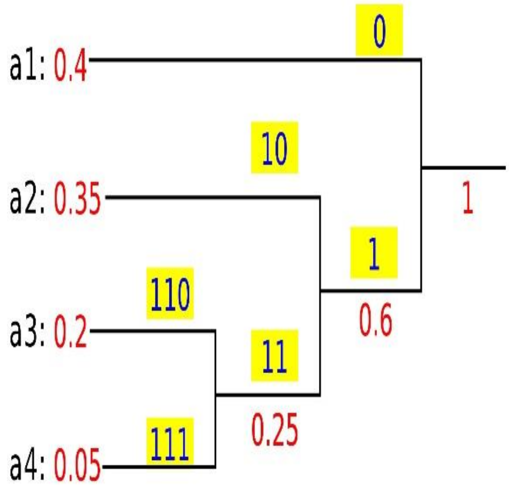

Fig 5: Huffman encoding example

1. Image Decompression

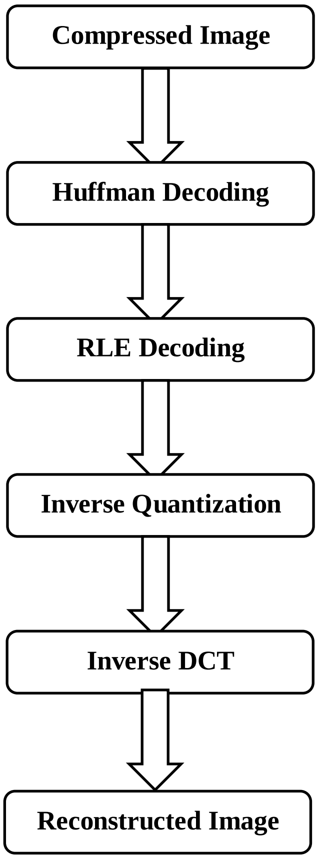

Fig 6: Flow chart of image decompression

In image decompression processes which are just reversed to compression are followed that is firstly Huffman code is decoded then RLE decoding takes place. After that Inverse Quantization and then inverse discrete cosine transform is performed. And that we get the reconstructed image that will be almost same as the original image but with a small data loss and compressed size. The reconstructed image will be looked as similar as the original image but with a few negligible changes that can be compensated in comparison to the compression ratio i.e. the size of decompressed image can be eight to ten times smaller than the original one. Out of the techniques used in image compression ,RLE and Huffman coding are lossless in nature whereas quantization is the part where the loss of image data occurs in the form of less colours and frequencies as compared to the input image but these losses are tolerable in front of high compression ratio.

II. EXPERIMENTAL RESULTS

The interpretation of work scheme is completed by performing 2D DCT at Lena's image with dimensions 256x256 in the JPEG format. Image compression and decompression steps are performed using different block sizes that is with block size 8x8, 16x16, 32x32. The variation in output image is compared on the basis of PSNR (stands for Peak Signal to Noise Ratio) and MSE (stands for Mean Square Error) and results are shown as:

Fig 7: Original Image (67438 bytes)

A. With Block Size - 8x8

Fig 8: Reconstructed Image (11588 bytes)

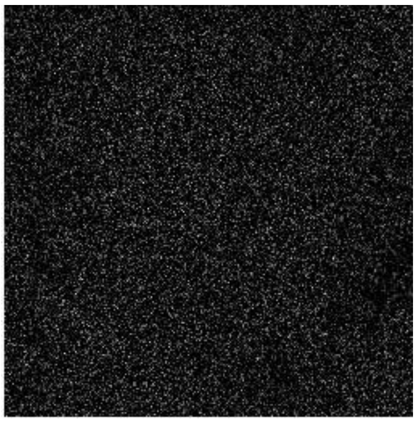

Fig 9: Error Image (8x8)

B. With Block Size – 16x16

Fig 10: Reconstructed Image (11573 bytes)

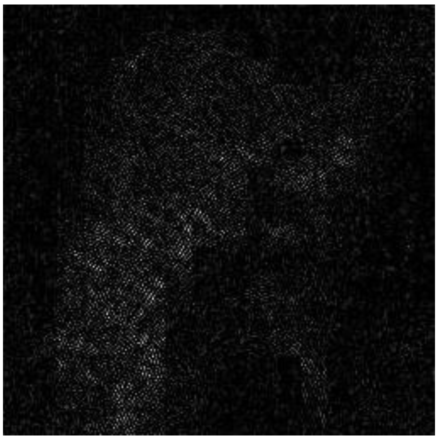

Fig 11: Error Image (16x16)

C. With Block Size – 32x32

Fig 12: Reconstructed Image (12069 bytes)

Fig 13: Error Image (32x32)

Block Size	**PSNR (in dB)**	**MSE**	**Output Size (in bytes)**	**Compression Ratio**
8x8	44.0536	2.5570	11588	5.820
16x16	36.1115	15.9196	11573	5.827
32x32	35.7295	17.3831	12069	5.588

Table 1: Comparison of output images with different block sizes

III. CONCLUSION

In the presented paper an efficient approach for image compression is proposed in which we use 2D DCT to compress the image efficiently. From table 1 we can conclude using the block size of 8x8 we get the best results in caparison to 16x16 or 32x32 block sizes. As PSNR is highest and MSE is lowest in case of 8x8 block size. From this we can say that 8x8 block size fulfils all the requirements of efficient and good quality image compression. Also if we use higher block size then CPU complexity is increased. Among these 3 block sizes CPU utilization will be maximum in case of 32x32 block size and will be minimum with block size 8x8.

IV. REFERENCES

[1]. Krikor L., Baba S., Arif T., and Shaaban Z., "Image Encryption using DCT and Stream Cipher", European Journal of Scientific Research, vol. 32, no. 1, pp.48-58,2009.

[2]. Ames G., "Image compression", International Journal of Advancements in Computing Technology, vol. 1, no. 2, pp. 205-220,2002.

[3]. C.S. Rawat and S. Meher, “A Hybrid Image Compression Scheme using DCT and Fractal Image Compression”, The International Arab Journal of Information Technology, vol. 10, noo. 6, Nov. 2013.

[4]. B. Santhanam, J.H. Mcclellan, “The discrete rotational fourier transform”, IEEE Trans. Signal Process, pp. 94-998, 1996.

[5]. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd edition, 2008.

[6]. N. Ahmed. T. Natarajan and K.R. Rao, “Discrete Cosine Transform”, IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, Jan 1974.

[7]. A. B. Watson, “Image Compression Using the Discrete Cosine Transform”, Mathematica Journal, vol. 4, no 1, pp. 81-88,1994.

[8]. G. K. Wallace, “The JPEG still picture compression standard”, Communications of ACM-Digital Multimedia Systems, vol. 34, no. 9, pp. 2174-2193, Sept 1992.

[9]. Robert M. Gray, IEEE, and David L. Neuhoff, IEEE “Quantization”, IEEE Trans. On Information Theory, Vol. 44, no. 6, pp. 2325-2383, October 1998.