International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-2, Issue-2, March-2014

Comparison of Image Compression Techniques: Huffman and DCT

Mr. Shirish V. Phakade, Mrs.Varsha K. Patil, Mr. Ajinkya Langade

Abstract-**The need for an efficient technique for compression of Images is ever increasing because the raw images need large amounts of disk space seems to be a big disadvantage during transmission and storage. Image compression is the application of data compression on digital images. Digital images contain large amount of digital information that need effective techniques for storing and transmitting large volume of data. Image compression techniques are used for reducing the amount of data required to represent a digital image**

This paper aims at the analysis of compression using DCT and Huffman transform by selecting proper better result for PSNR, compression ratio, RMSE have been obtained.In this paper we proposed the lossless method of image compression and decompression using a simple coding technique called Huffman coding. This technique is simple in implementation and utilizes less memory.

A software algorithm has been developed and implemented to compress and decompress the given image using Huffman coding techniques in a MATLAB platform. An Image can be compressed with use of Discrete Cosine Transformation (DCT), quantization encoding are the steps in the compression of the JPEG image format. The 2-D Discrete Cosine transform is used to convert the 8×8 blocks of image into elementary frequency components .The frequency components(DC and AC) are reduced to zero during the process of quantization which is a lossy process .The quantized frequency components are coded into variable length code words using encoding process. Distortion between the original image and reconstructed image is measured with PSNR (peak signal to noise ratio) with different compression factors. The compression ratio and PSNR values are different for different images.

Index Terms- **DCT, PSNR**

I. INTRODUCTION

A digital image obtained by sampling and quantizing a continuous tone picture requires an enormous storage. For instance, a 24 bit color image with 512x512 pixels will occupy 768 Kbyte storage on a disk, and a picture twice of this size will not fit in a single floppy disk. To transmit such an image over a 28.8 Kbps modem would take almost 4 minutes. There are different techniques for compressing images. They are broadly classified into two classes called lossless and lossy compression techniques[4].

It's well known that the Huffman's algorithm is generating minimum redundancy codes compared to other algorithms. The Huffman coding has effectively used in text, image, video compression, and conferencing system such as, JPEG, MPEG-2, MPEG-4, and H.263 etc.. The Huffman coding technique collects unique symbols from the source image and calculates its probability value for each symbol and sorts the symbols based on its probability value [4].

The JPEG process is widely used form of Lossy image, compression that centers around the Discrete Cosine Transform (DCT). The DCT works by separating images into parts differing frequencies. During a step called Quantization. Where part of compression actually occurs, the less important frequencies are discarded, hence the use of term "Lossy". Then, only the most important frequencies that remain are used to retrieve the image in the decompression process. As a result reconstructed images contains some distortion ; but as we shall soon see, this level of distortion can be adjusted during the Compression stage. The JPEG method is used for both color and black-and-white images[12].

II. NEED FOR COMPRESSION

The following example illustrates the need for compression of digital images.

a. To store a color image of a moderate size, e.g. 512×512 pixels, one needs 0.75 MB of disk space.

b. A 35mm digital slide with a resolution of 12µm requires 18 MB.

c. One second of digital PAL (Phase Alternation Line) video requires 27 MB.

To store these images, and make them available over network (e.g. the internet), compression techniques are needed. Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. According to mathematical point of view, this amounts to transforming a two-dimensional pixel array into a statistically

Manuscript Received March 21, 2014

Mr. Shirish V. Phakade, Electronics and Telecommunication department, PVPIT, Budhgaon,Sangli, M.S., India,7798025050., (e-mail: phakadesv@gamil.com).

Mrs. Varsha K. Patil, Electronics and Telecommunication department, AIMSS's IOIT ,Pune,M.S.,India,9325643275 (e-mail: varshapatil101@gmail.com).

Mr.Ajinkya Langde, Electronics and Telecommunication department, PVPIT, Budhgaon,Sangli, M.S., India,8600765874 , (e-mail: Ajinkya.langade57@gmail.com).

uncorrelated data set. The transformation is applied prior to storage or transmission of the image. At receiver, the compressed image is decompressed to reconstruct the original image or an approximation to it [4].

III. TYPES OF COMPRESSION

Compression can be divided into two categories, as Lossless and Lossy compression. In lossless compression, the reconstructed image after compression is numerically identical to the original image. In lossy compression scheme, the reconstructed image contains degradation relative to the original. Lossy technique causes image quality degradation in each compression or decompression step. In general, lossy techniques provide for greater compression ratios than lossless techniques.
The following are the some of the lossless and lossy data compression techniques [4]:

1 Lossless coding techniques

a. Run length encoding
b. Huffman encoding
c. Arithmetic encoding
d. Entropy coding
e. Area coding

2 Lossy coding techniques

a. Predictive coding
b. Transform coding (FT/DCT/Wavelets)

IV. WORKING OF IMAGE COMPRESSION TECHNIQUES:

A. *Huffman coding-*

Huffman code procedure is based on the two observations.
a. More frequently occurred symbols will have shorter code words than symbol that occur less frequently.
b. The two symbols that occur least frequently will have the same length.

Table 1: Huffman source reduction.

Original source		Source reduction			
S	P	1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

S-source, P-probability

Table 2 : Huffman Code Assignment Procedure

Original source		Source reduction			
S	P	1	2	3	4
a2	0.4[1]	0.4[1]	0.4[1]	0.4[1]	0.6[0]
a6	0.3[00]	0.3[00]	0.3[00]	0.3[00]	0.4[1]
a1	0.1[011]	0.1[011]	0.2[010]	0.3[01]	
a4	0.1[0100]	0.1[0100]	0.1[011]		
a3	0.06[01010]	0.1[0101]			
a5	0.04[01011]				

S-source, P-probability

At the far left of the table I the symbols are listed and corresponding symbol probabilities are arranged in decreasing order and now the least two probabilities are merged as here 0.06 and 0.04 are merged, this gives a compound symbol with probability 0.1, and the compound symbol probability is placed in source reduction column1 such that again the probabilities should be in decreasing order. so this process is continued until only two probabilities are left at the far right shown in the above table as 0.6 and 0.4. The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to its original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As shown in table II these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and would work just and well). As the reduced source symbol with probabilities 0.6 was generated by combining two symbols in the reduced source to, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrary appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original course is reached. The final code appears at the far-left in table 1.8. The average length of the code is given by the average of the product of probability of the symbol and number of bits used to encode it. This is calculated below [4]:

Lavg = (0.4)(1) +(0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5) = 2.2 bits/ symbol

and the entropy of the source is 2.14bits/symbol, the resulting Huffman code efficiency is2.14/2.2 = 0.973.

Entropy, $H=-\sum P(aj)\log P(aj)$

Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time.

B. *Huffman decoding:*

After the code has been created, coding and/or decoding is accomplished in a simple look-up table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code, because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each codeword in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. For the binary code of table 3.2, a left-to-right scan of then coded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a3.The next valid code is 011, which corresponds to symbola1. Valid code for the symbol a2 is 1,valid code for the symbols a6

is 00,valid code for the symbol a6 is Continuing in this manner reveals the completely decoded message a5 a2 a6 a4 a3 a1 , so in this manner the original image or data can be decompressed using Huffman decoding as explained above [4].

At first we have as much as the compressor does a probability distribution. The compressor made a code table. The decompressor doesn't use this method though. It instead keeps the whole Huffman binary tree, and of course a pointer to the root to do the recursion process. In our implementation we'll make the tree as usual and then you'll store a pointer to last node in the list, which is the root. Then the process can start. We'll navigate the tree by using the pointers to the children that each node has. This process is done by a recursive function which accepts as a parameter a pointer to the current node, and returns the symbol[4].

C. DCT compression/decompression-

Image compression is a technique used to reduce the storage and transmission costs. The existing techniques used for compressing image files are broadly classified into two categories, namely lossless and lossy compression techniques. In lossy compression techniques, the original digital image is usually transformed through an invertible linear transform into another domain, where it is highly de-correlated by the transform. This de-correlation concentrates the important image information into a more compact form. The transformed coefficients are then quantized yielding bit-streams containing long stretches of zeros. Such bit-streams can be coded efficiently to remove the redundancy and store it into a compressed file. The decompression reverses this process to produce the recovered image [10].

The 2-D discrete cosine transform (DCT) is an invertible linear transform and is widely used in many practical image compression systems because of its compression performance and computational efficiency [10]. DCT converts data (image pixels) into sets of frequencies. The first frequencies in the set are the most meaningful; the latter, the least. The least meaningful frequencies can be stripped away based on allowable resolution loss. DCT-based image compression relies on two techniques to reduce data required to represent the image. The first is quantization of the image's DCT coefficients; the second is entropy coding of the quantized coefficients. Quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it. Quantization is a lossy process and implies in a reduction of the color information associated with each pixel in the image. Entropy coding is a technique for representing the quantized coefficients as compactly as possible [10].

V. IMPLEMENTATION OF HUFFMAN CODING, DECODING AND DCT

A. Huffman Coding and Decoding Algorithm:

1- Read the image on to the workspace of the matlab.
2- Call a function which will find the symbols (i.e. pixel value which is non-repeated).
3- Call a function which will calculate the probability of each symbol.
4- Probability of symbols are arranged in decreasing order and lower probabilities are merged and this step is continued until only two probabilities are left and codes are assigned according to rule that: the highest probable symbol will have a shorter length code.
5- Further Huffman encoding is performed i.e. mapping of the code words to the corresponding symbols will result in a compressed data.
6- The original image is reconstructed i.e. decompression is done by using Huffman decoding[4].

B. DCT Coding and Decoding Algorithm

The following is a general overview of the JPEG process.

1.The image is broken into 8x8 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. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.

5.When desired, the image is reconstructed through decompression, a process that uses the inverse Discrete Cosine Transform (IDCT)[12].

VI . RESULTS

A. DCT result

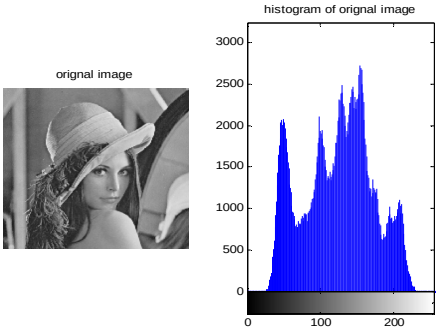

Fig.1 Original image

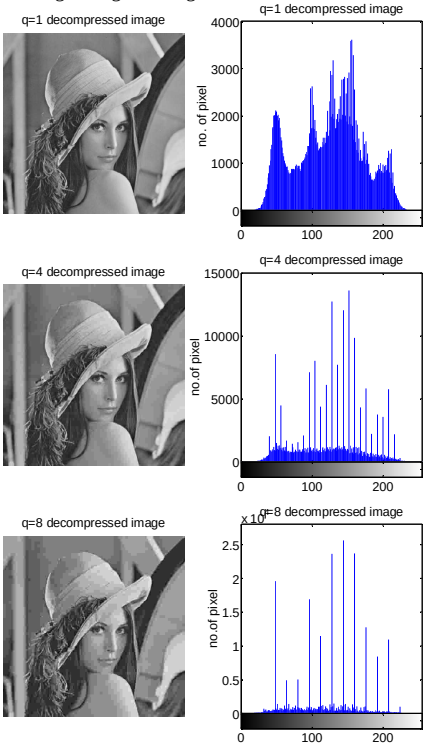

Fig.2 Decompressed images with varied "q"

B, Huffman result

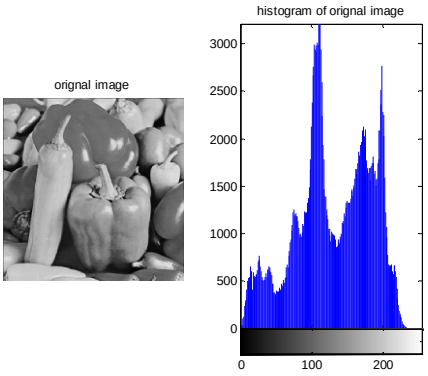

Fig.3 Original image

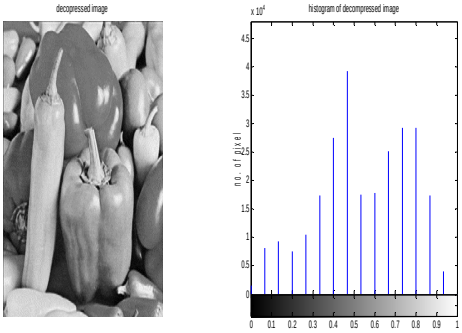

Fig.3 Decompressed image

VII. COMPARISON TABLE

TABLE 3a: HUFFMAN AND DCT PERFORMANCE COMPARISON

Techniques	Image	CR	PSNR	RMSE
Huffman	Pepper	4.13	31.73	6.56
DCT(q=1)	Leena	11.2847	35.785	4.142
DCT(q=4)	Leena	28.3706	31.245	6.987
DCT(q=8)	Leena	47.1143	28.452	9.636

Table 3b: Huffman And Dct Performance Comparison With Timie Details

Huffman Result:

Image	CR	PSNR	RMSE	Encoding time	Decoding time
Lena512.bmp	3.877	31.9053	6.4755	0.838035 sec	1.691683 sec
Pout.tif	4.4046	31.9905	6.4123	0.079782 sec	0.260826 sec
Peppers.bmp	4.1311	31.7886	6.5631	0.911086 sec	1.298329 sec

DCT Result:

Image	CR	PSNR	RMSE	Encoding time	Decoding time
Lena512(Q=1)	11.2847	35.7850	4.1427	2.280217 sec	2.089436 sec
Lena512(Q=2)	17.8014	33.6906	5.2724	2.170582 sec	2.030948 sec
Lena512(Q=4)	28.3706	31.2450	6.9870	2.144558 sec	2.017098 sec
Lena512(Q=8)	47.1143	28.4527	9.6362	2.128752 sec	2.020122 sec
Lena512(Q=12)	64.0939	26.6041	11.9216	2.111116 sec	2.000551 sec
Pout(Q=1)	15.6873	40.0181	2.5447	0.688853 sec	0.607514 sec
Pout(Q=2)	25.3411	37.5083	3.3972	0.606843 sec	0.560530 sec
Pout(Q=4)	43.4328	34.3840	4.8679	0.581925 sec	0.549482 sec
Pout(Q=8)	77.2566	30.8553	7.3076	0.578286 sec	0.544563 sec
Pout(Q=12)	114.8684	28.6379	9.4329	0.587155 sec	0.543685 sec
Peppers(Q=1)	12.8150	37.4272	3.4291	2.279321 sec	2.108851 sec
Peppers(Q=2)	30.0969	33.9209	5.1345	2.188939 sec	2.073299 sec
Peppers(Q=4)	30.0969	33.9209	5.1345	2.210023 sec	2.077152 sec
Peppers(Q=8)	46.7780	29.8994	8.1578	2.210023 sec	2.077152 sec
Peppers(Q=12)	61.8848	27.5519	10.6892	2.185994 sec	2.079803 sec

By observing table we conclude that in Huffman compression PSNR, RMSE and compression ratio for the all images are nearly same. But in DCT as quality factor increases compression ratio goes on increasing, PSNR goes on decreasing and RMSE goes on increasing. We can obtain better quality with better compression till quality factor 8. The Results are observed for three different images as listed in table.

VIII. CONCLUSION

In this Project, we have considered that DCT and Huffman coding for image compression and decompression. By considering several images as inputs, it is observed that RMSE is low and PSNR is high in Huffman than DCT based compression.

From the results it is concluded that overall performance of Huffman is better than DCT on the basis of compression rates. In DISCRETE COSINE TRANSFORM image need to be “blocked”, correlation across the block boundaries is not eliminated. This results in noticeable and annoying “blocking artifact” particularly at low bit rates. Wavelets are good to represent the point singularities and it cannot represent line singularities.

REFERENCES

[1] A.B.Watson,“Image Compression using the DCT” , Mathematic Journal, 1995,pp.81-88.
[2] D.A.Huffman, A Method for the construction of Minimum-redundancy Codes, Proc. IRE, vol.40, no.10, pp.1098-1101,1952.
[3] Ternary Tree & FGK Huffman Coding Technique Dr. Pushpa R.Suri † and Madhu Goel Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, India .
[4] A New Lossless Method Of Image Compression And Decompression Using Huffman Coding Techniques Jagadish H. Pujar, Lohit M. Kadlaskar.
[5] A Comparative Study Of Image Compression Method Ashwin Swaminathan, Gaurav Agarwal.
[6] Improvement In Compression Efficiency Of Huffman Coding Mohd. Faisal Muqtida, Raju Singh Kushwaha Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science.
[7] RL-Huffman Encoding for Test Compression and Power Reduction in Scan Applications-MEHRDAD NOURANI and MOHAMMAD HITEHRANIPOUR, The University of Texas at Dallas.
[8] Efficient Huffman decoding by MANOJ AGGRAWAL and AJAI NARAYAN.
[9] Analysis of Image Compression Techniques using DCT
. VirenderPoswal * , Dr.Priyanka **
*ECE Deptt. D.C.R.U.S.T, Murthal, Haryana,**Asso.Prof. ECE Deptt. D.C.R.U.S.T, Murthal, Haryana
[10] Analysis of Image Compression Algorithm using DCT Maneesha Gupta, Dr.Amit Kumar Garg
[11] Image Compression Using DCT and Prabhakar.Telagarapu, V.Jagan Naveen, A.Lakshmi..Prasanthi, G.Vijaya Santhi. Wavelet Transformations
[12] Image compression and the DCTKen cabeen and Peter Gent, math45,college of the Redwoods.