National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2012)
Proceedings published by International Journal of Computer Applications® (IJCA)

Lossy Image Compression using Discrete Cosine Transform

Pravin B. Pokle
J.L.Chaturvedi College of Engg
746, Nandanvan, Nagpur

N. G. Bawane
S.B.Jain Institute of Engg.
Katol Road, Nagpur

ABSTRACT

In recent times the integration of video, audio and data in telecommunication devices has revolutionized the world communication. It has proven to be useful to almost every industry: the corporate world, entertainment industry, multimedia, education and even many household domestic appliances.

The major problems encountered with these applications are the high data rates, high bandwidth and large memory required for storage and computing resources. Even with faster internet, throughput rates and improved network infrastructure, there are major bottlenecks in transferring such high volume data through the network due to bandwidth limitations. This justifies the need to develop compression techniques in order to make the best use of available bandwidth [1].

This paper presents how the digital image is compressed using discrete cosine transform and the comparative study with other methods.
Keywords: JPEG-Joint Photographic Experts Group, DCT-discrete cosine transform, FFT-Fast Fourier Transform, IDCT-Inverse Discrete Cosine Transform.

1. INTRODUCTION

In 1992, the Joint Photographic Experts Group (JPEG) released their first international compression standard for continuous-tone still Images, both in colour and gray scale. Fig.1 illustrates a typical image compression system with three blocks to implement the basic operations [2].

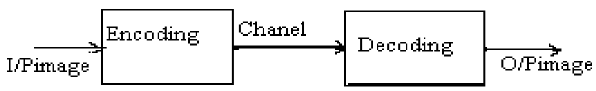


Fig. 1 Generalized Image compression system.

The input image is initially divided into a number of blocks or frames prior to transformation. The coefficients obtained after image transformation are quantized and later entropy encoded to yield the final compressed image. The signal processing algorithms used in the transformation block.

2. TYPES OF COMPRESSION

Two types of compression techniques exist, lossless and lossy. In lossless compression, the reconstructed image after compression is numerically identical to the original image on a pixel by-pixel basis. However, only a modest amount of Compression is achievable in this technique. In lossy compression, the reconstructed image contains degradation relative to the original, because redundant information is discarded during compression. As a result, much higher compression is achievable, and under normal viewing conditions, no visible loss is perceived. Most of the lossy compression techniques use discrete cosine transform.

3. DISCRETE COSINE TRANSFORM

In today's technological world, we need for efficient ways of storing large amounts of data i.e. more memory requirement and due to the bandwidth and storage limitations, images must be compressed before transmission and storage.
However, the compression will reduce the image fidelity, especially when the images are compressed at lower bit rates. The reconstructed images suffer from blocking artifacts and the image quality will be severely degraded under the circumstance of high compression ratios. In order to have a good compression ratio without losing too much of information when the image is decompressed we use DCT.

A discrete cosine transform (DCT) is basically a 'lossy' image compression technique which uses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies that centres on the Discrete Cosine Transform. Here a reversible linear transform such as Fourier Transform is used to map the image into a set of transform coefficients, which are then Quantized and encoded. DCT and Fourier transforms convert images from time-domain to frequency-domain to de-correlate pixels [3].

3.1 TYPES OF DCT

There are two types of DCT as given below [4].

3.1.1 One-Dimensional DCT

The most common DCT of a 1-D sequence of length N is

$$P(i) = \alpha(i) \sum_{x=0}^{L-1} f(x) \cos\left[\frac{\pi(2x+1)i}{2L}\right] \qquad (1)$$

For i = 0, 1, 2… L −1.

and the inverse transformation is defined as

$$f(x) = \sum_{i=0}^{L-1} \alpha(i) P(i) \cos\left[\frac{\pi(2x+1)i}{2L}\right] \qquad (2)$$

For x = 0, 1, 2… L −1.

Where α (i) is defined as

$$\alpha(i) = \begin{cases} \sqrt{\frac{1}{L}} & for\ i = 0 \\ \sqrt{\frac{2}{L}} & for\ i \neq 0 \end{cases} \qquad (3)$$

From first equation it is clear that for i = 0, P (i = 0) = $\sqrt{\frac{1}{L}}\sum_{x=0}^{L-1} f(x)$. (4)

Thus, the first transform coefficient is the average value of the sample sequence. In this paper, this value is referred to as the DC Coefficient. All other transform coefficients are called the AC Coefficients.

3.1.2 Two-Dimensional DCT

The Discrete Cosine Transform (DCT) is one of many transforms that takes its input and transforms it into a linear combination of weighted basis functions i.e. the frequency. The 2-D Discrete Cosine Transform is just a one dimensional DCT applied twice, once in the x direction, and again in the y direction. One can imagine the computational complexity of doing so for a large image. Thus, many algorithms, such as the Fast Fourier Transform (FFT), have been created to speed the computation.

The DCT equation (Eq.5) computes the i, j^{th} entry of the DCT of an image.

$$Q(l,j) = \frac{1}{\sqrt{2L}} P(l)P(j) \sum_{x=0}^{L-1}\sum_{y=0}^{L-1} T(x,y) \cos\left[\frac{(2x+1)l\pi}{2L}\right] * \cos\left[\frac{(2y+1)j\pi}{2L}\right] \quad (5)$$

$$P(i) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ i = 0 \\ 1, & if\ i > 0 \end{cases} \quad (6)$$

Where T(x, y) is the x, y^{th} element of the image represented by the matrix T. L is the block size on which the DCT is applied. The equation calculates one entry (l, j th) of the transformed image from the pixel values of the original image matrix. For the standard 8x8 block that JPEG compression uses, L equals 8 and x and y range from 0 to 7. Thus Q(l, j) would be given by equation (7).

$$Q(i,j) = \frac{1}{4} P(i)P(j) \sum_{i=0}^{7}\sum_{j=0}^{7} T(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] * \left[\frac{(2y+1)j\pi}{16}\right] \quad (7)$$

Because the DCT uses cosine functions, the resulting matrix depends on the horizontal and vertical frequencies. Therefore an image block with a lot of change in frequency has a very random looking resulting matrix, while an image matrix of just one color, has a resulting matrix of a large value for the first element and zeroes for the other elements.

4. THE JPEG PROCESS

This section describes the JPEG process. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. JPEG typically achieves 10:1 compression with little perceptible loss in image quality [5, 6]. The JPEG process is carried out in five steps as follows:

4.1 Break the image into 8x8 blocks of pixels.

To get the matrix form of Equation (5), we will use the following equation,

$$Uij = \begin{cases} \frac{1}{\sqrt{L}}, & if\ l = \\ \sqrt{\frac{2}{L}} \cos\left[\frac{(2j+1)l\pi}{2L}\right], & if\ l > \end{cases} \quad (8)$$

For an 8x8 block it results in this matrix:
The first row (i : 1) of the matrix has all the entries equal to 1/8 as expected from Equation (8).The columns of U form an orthonormal set, so U is an orthogonal matrix. When doing the inverse DCT the orthogonality of U is important, as the inverse of U is U' which is easy to calculate.

4.2 Apply DCT to each block from left to right, top to bottom.

The pixel values of a black-and-white image range from 0 to 255 in steps of 1, where pure black is represented by 0, and pure white by 255. Thus it can be seen how a photo, illustration, etc. can be accurately represented by these 256 shades of gray. Since an image comprises hundreds or even thousands of 8x8 blocks of pixels, Now, start with a block of image pixel values. This particular block was chosen from the very upper- left-hand corner of an image. Because the DCT is designed to work on pixel values ranging from -128 to 127, the original block is "leveled off" by subtracting 128 from each entry. This results in the following matrix.
Now perform the Discrete Cosine Transform, which is simply done by matrix multiplication.

D=TMT' (9)

In Equation (9), 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 matrix.
If this matrix, be c (i, j), where i and j range from 0 to 7. The top-left coefficient, c(0, 0), correlates to the low frequencies of the original image block. As we move away from c(0,0) in all directions, the DCT coefficients correlate to higher and higher frequencies of the image block, where c(7,7) corresponds to highest frequency. Higher frequencies are mainly represented as lower number and Lower frequencies as higher number. It is important to know that human eye is most sensitive to lower frequencies.

4.3 Quantization to compress each block.

Quantization is achieved by dividing each element in the transformed image matrix 'D' by corresponding element in the quantization matrix, and then rounding to the nearest integer value. Subjective experiments involving the human visual system have resulted in the JPEG standard quantization matrix. Considering a quality level, this matrix renders both high compression and excellent decompressed image quality.

The scaled quantization matrix is then rounded and clipped to have positive integer values ranging from 1 to 255.
For rounded off we use equation (10)

$$C_{ij} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right) \quad (10)$$

Thus we get the new matrix where the coefficients situated near the upper-left corner correspond to the lower frequencies to which the human eye is most sensitive of the image block. Whereas, zeros represent less important, higher frequencies that have been discarded, giving rise to the lossy part of compression. Thus, only nonzero coefficients will be used to reconstruct the image.

4.4 The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
The quantized matrix after rounded off is now ready for the final step of compression. Before storage, all coefficients of

4.5 Decompression using Inverse Discrete Cosine Transform (IDCT).
Reconstruction of image by decoding the bit stream representing the Quantized matrix. Each element of this matrix is then multiplied by the corresponding element of the quantization matrix originally used as under:

$$R_{i,j} = Q_{i,j} \times C_{i,j} \qquad (11)$$

The IDCT is next applied to matrix R, which is rounded to the nearest integer. Finally, 128 is added to each element of that result, giving us the decompressed JPEG version N of our original 8x8 image block M.

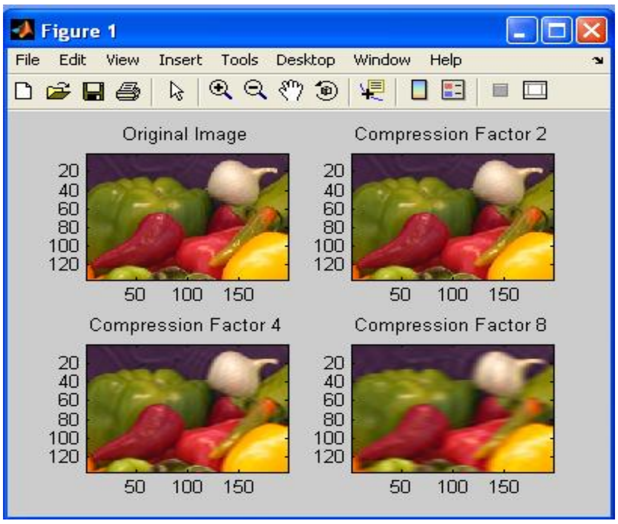
$$N = \text{round}\,(T'RT) + 128 \qquad (12)$$

Thus we get decompressed matrix as under:

5. ADVANTAGES OF DCT OVER OTHER TRANSFORM

1. DCT is the most used transform for compression because its energy compaction is better than the other transforms.
2. Because of availability of fast algorithms it is computationally simpler.
A major problem related with the DCT techniques is that the decoded images, especially at low bit rates, exhibit visible this matrix are converted by an encoder to a stream of binary data (01101011...).

6. RESULTS AND CONCLUSION



In the above figure, first figure shows the original image, and the others represents the compressed images with different compression factor. It is seen that increasing compression factor will degrade the quality of image.
gray-level discontinuities along the block boundaries. This is because of discarding the similar pixels. In order to overcome this drawback, a new concept based on wavelets is used.

From the above two figures and matrix, if we compare the original matrix with decompressed matrix, nearly 70% of the DCT coefficients were discarded prior to image block decompression/reconstruction. Also from MATLAB algorithm even by changing compression factor of 8, the retrieved image is looking similar to the original one. DCT takes advantage of redundancies in the data by grouping pixels with similar frequencies together and moreover if we observe as the resolution of the image is very high, even after sufficient compression and decompression there is very less change in the original and decompressed image. Thus, we conclude at the same compression ratio the difference between original and decompressed image goes on decreasing as there is increase in image resolution. Further research work is in progress to improve the image compression using DCT and comparing its performance with other existing techniques.

6. REFERENCES

[1] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", 2nd Ed.,Prentice Hall, 2004.

[2] A. Said and W. A. Peralman, "An Image Multiresolution Representation for Lossless and Lossy Compression," IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 5, NO. 9, SEPTEMBER 1996.

[3] I-Ming Pao and Ming-Ting Sun, Fellow, “Modeling DCT Coefficients For Fast Video Encoding”, IEEE Transactions on circuits and systems For video tech. VOL. 9, NO. 4, JUNE 1999

[4] E. Lam, J. A. Goodman, “A Mathematical Analysis of the DCT Coefficient Distributions for Images” in IEEE Trans. on Image Processing 9(10), 1661-1666 (2000).

[5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 Still Image Compression Standard,” IEEE Signal Processing Magazine, pp. 36-58, September 2001.

[6] Shantanu D. Rane and Guillermo Sapiro, Member, IEEE,”Evaluation of JPEG-LS, the New Lossless and Controlled-Lossy Still Image Compression Standard, for Compression of High-Resolution Elevation Data”, IEEE Transactions on Geoscience and Remote sensing, VOL. 39, NO. 10, Oct. 2001

[7] Matteo Frigo and Steven G. Johnson: FFTW, "The Design and Implementation of FFTW3”, Proceedings of the IEEE 93 (2), 216–231 (2005).