International Journal of Recent Advances in Multidisciplinary Topics
Volume 2, Issue 6, June 2021
https://www.ijramt.com | ISSN (Online): 2582-7839

Digital Image Compression using DCT and SVD

Jerry Lalnunhlima[1*], T. D. Johnson[2], Lalruatzuala[3]
[1,2,3]*Student, Department of Computer Engineering, Mizoram University, Aizawl, India*

***Abstract*: In our digital world, almost everybody cannot live without electronic devices i.e. mobile phones, computers, cameras, etc. As it is used in most of our daily lives i.e. jobs, rail stations, grocery stores, airports, etc. It plays a crucial role in our daily needs, and almost our images are stored as digital images in our device memory. So we need to have a large storage space for our data. So, image compression plays an important role in making our data size smaller by removing unnecessary objects (pixels or noises) in the images so as to maintain a larger amount of storage. Image compression is the method of data compression on digital images. The larger the data files associated with the images, thus the drives need for high compression ratios to make storage. For example, if we want to store larger files or videos (particularly movies) to our device memory, as each frame of a video is made of an image; thus the drives need to have high compression ratios in order to stores/transmits the data in an efficient form. Image compression process uses two techniques to compress an image- Lossy image compression- to achieve much higher compression, and Lossless image compression- to obtain zero quality loss images after compression. Lossy image compression techniques have much higher compression ratios than Lossless image compression. This paper describes Lossy compression techniques using Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) and deals with the comparative study of compressed images on the basis of different values.**

***Keywords*: Discrete cosine transform, image compression, singular value decomposition.**

1. Introduction

Image Compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission. Image compression is minimizing the size in bytes of a graphic file without degrading the quality of the image to an unacceptable level. It involves minimization of the number of information carrying units, pixels. This means that an image where adjacent pixels have almost the same values leads to spatial redundancy. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the internet or downloaded from web pages. In this paper, we review the different image compression techniques/methods that are used to implement a compression in Discrete Cosine Transform (DCT) and Singular Value Decomposition (SVD) using MATLAB.

2. Types of Image Compression

Image compression process uses two techniques to compress an image-Lossy image compression and Lossless image compression.

A. *Lossy Image Compression*

Lossy image compression is a type of compression where a certain amount of information is discarded which means that some data are lost and hence the image cannot be decompressed with 100% originality. Lossy image compression schemes are capable of achieving much higher compression. Lossy methods are especially suitable for natural images such as photos in applications where minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

B. *Lossless Image Compression*

Lossless image compression is a compression algorithm that allows the original image to be perfectly reconstructed from the original data. In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. Lossless compression is preferred for archival purposes and often imaging, technical drawings, clip art or comics.

3. Methodology

A. *Digital Image Compression using Discrete Cosine Transform (DCT)*

For compression purpose, the higher the capability of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) have become the most widely used transform coding techniques. DCT convert an image into its equivalent frequency domain by partitioning image pixel matrix into blocks of size N*N. An image is a 2D pixel matrix hence 2D DCT is used to transform an image. To perform DCT transformation on an image, first we have to fetch image file information (pixel value in term of integer having range 0-255) which we divide in block of 8 X 8 matrix and then we apply DCT on that block of data.

*Corresponding author: jlalnunhlima@gmail.com

DCT is applied to each block by multiplying the modified block with DCT matrix on the left and transpose of DCT matrix on its right. DCT coding algorithms usually start by partitioning the original image into sub images (blocks) of small size (usually 8 × 8). For each block the transform coefficients are calculated, effectively converting the original 8 × 8 array of pixel values into an array of coefficients within which the coefficients closer to the top-left corner usually contain most of the information needed to quantize and encode (and eventually perform the reverse process at the decoder's side) the image with little perceptual distortion. The resulting coefficients are then quantized and the output of the quantize is used by symbol encoding techniques to produce the output bit stream representing the encoded image. DCT separates images into parts of different frequencies where less important frequencies are discarded through quantization and important frequencies are used to retrieve the image during decompression. Compared to other input dependent transforms, DCT has many advantages

1. It has been implemented in single integrated circuit;
2. It has the ability to pack most information in fewest coefficients;
3. It minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible.

The forward 2D_DCT transformation is given by the following equation:

$$C(u,v)=D(u)D(v) \sum_{x=0}^{N-1} f(x,y)\cos[(2x+1)u\pi/2N] \cos[(2y+1)v\pi/2N]$$

Where, u,v=0,1,2,3,…......................... ,N-1 (Equation 1).

The inverse 2D-DCT transformation is given by the following equation:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} D(u)D(v)D(u,v)\cos[(2x+1)u\pi/2N]x \cos(2y+1)v\pi/2N$$

] Where

$D(u)=(1/N)$ ^1/2

for u=0 D(u)=2(/N)^1/;

for u=1,2,3…….,(N-1).

The f(x,y) is the value of each pixel in the selected 8×8 block, and the D(u,v) is the DCT coefficient after transformation. The transformation of the 8×8 block is alsoa 8×8 block composed of D(u,v). After transformation, the element in the upper most left corresponding to zero frequency in both directions is the "DC coefficient" and the rest are called "AC coefficients."

B. JPEG Process

1. Original image is divided into blocks of 8 x 8.
2. Pixel values of a black and white image range from 0-255 but DCT is designed to work on pixel values ranging from -128 to 127. Therefore each block is modified to work in the range.
3. Equation (1) is used to calculate DCT matrix.
4. DCT is applied to each block by multiplying the modified block with DCT matrix on the left and transpose of DCT matrix on its right.
5. Each block is then compressed through quantization.
6. Quantized matrix is then entropy encoded.
7. Compressed image is reconstructed through reverse process.
8. Inverse DCT is used for decompression.

C. Compression Model

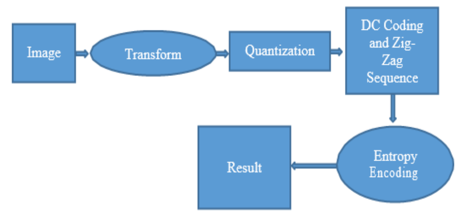

Fig. 1. JPEG Compression Model

D. Proposed Work

1) Color Space Transformation

Colors in an image is usually represented through RGB space. First step in the compression is to represent these colors through another color space YCb Cr, where Y represents luminosity component which essentially represents brightness or intensity of image pixels. Cb and Cr represent chrominance (or color) of blue and red respectively. The intensity of each component is represented by a number in the range [0, 255]. This transformation is done to achieve high compression. So, in this step we have Y, Cb, Cr channels in place of RGB.

2) Transform

This stage is necessary to convert the image data from spatial domain into frequency domain. This transform converts the image data into a form that can be easily compressed. The transform that is used is Discrete Cosine Transform (DCT). To perform DCT transformation on an image, first we have to fetch image file information (pixel value in term of integer having range 0-255) which we divide in block of 8 X 8 matrix and then we apply DCT on that block of data. Pixel values of a black and white image range from 0- 255 but DCT is designed to work on pixel values ranging from -128 to 127. Therefore each block is modified to work in the range.

3) Quantization

Quantization is a process of reducing number of bits needed to store data of high frequency by reducing precision of its values. The goal of quantization is to reduce most of the less important high frequency DCT coefficients to zero, the more zeroes we generate the better the image will compress. The matrix that we get after DCT step contains all important data at the top left corner and all A.C. signals at lower right corner. JPEG provides us with a quantization matrix. In JPEG standards, to get quantized matrix every element in DCT matrix is divided by corresponding element in quantization matrix. The quantization matrix is designed so as to contain higher value at lower right side so as to give lower values when element of DCT matrix is divided by its corresponding element.

Apply the following formula:

(i, j) = Round [M(i, j)/Q(i, j)]

Where, B(i, j) denotes element of new quantized matrix, M(i, j) denotes element of DCT matrix, and Q(i, j) denotes element of quantization matrix.

After dividing elements of DCT matrix by corresponding element of quantization matrix, we round off the value to its nearest integer. Thus rounding off most values to 0.

4) DC Coding and Zig-Zag Sequence

After quantization, the DC coefficient is treated separately from the 63 AC coefficients. The DC coefficient is a measure of the average value of the original 64 image samples. Because there is usually strong correlation between the DC coefficients of adjacent 8×8 blocks, the quantized DC coefficient is encoded as the difference from the DC term of the previous block. This special treatment is worthwhile, as DC coefficients frequently contain a significant fraction of the total image energy. The other 63 entries are the AC components. They are treated separately from the DC coefficients in the entropy coding process.

The Zig-Zag Scan order:

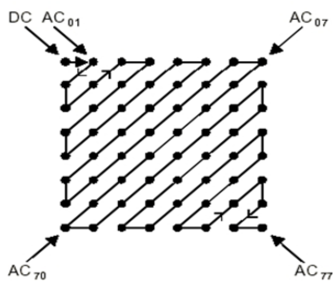

Fig. 2. DC Coding and Zig-Zag Scanning

5) Entropy Encoding

Entropy encoding is a lossless form of storing data where redundant data is stored in fewer bits and non-redundant data is represented in many bits. The actual Run Length coding of JPEG is applied Zig-Zag on a matrix and is of format,

(Run-Length, Size) (Amplitude)

Where Run-Length = the no. of zeros before non-zero element, x

Size = No. of bits required to represent non-zero element, x

Amplitude = Bit representation of x.

E. Digital Image Compression using Singular Value Decomposition (SVD)

Using SVD for image compression, we can decompose a given image into the three color channels red, green and blue. Each channel can be represented as a (m × n) - matrix with values ranging from 0 to 255. We will now compress the matrix a representing one of the channels. To do this, we compute an approximation to the matrix a takes only a fraction of the space to store. Now here's the great thing about SVD: the data in the matrices U, Σ and V is sorted by how much it contributes to the matrix A in the product. That enables us to get quite a good approximation by simply using only the most important parts of the matrices. We now choose a number k of singular values that we are going to use for the approximation. The higher this number, the better the quality of the approximation gets but also the more data is needed to encode it. We now take only the first k columns of U and V and the upper left (k × k)-square of Σ, containing the k largest (and therefore most important) singular values. We then have

$$A \approx U_k \Sigma_k V_k^T$$

The amount of data needed to store this approximation is proportional to the colored area:

Compressed Size = m × k + k + k × n = k × (1 + m + n) (Actually, slightly less space is needed due to the orthogonality of U and V.) One can prove that this approximation is optimal in a certain sense.

4. Result

In this section, we display the various results of our numerical implementation of the above procedure in Matlab.

Original Image (174 KB)

DCT Compressed Image (54 KB)

Fig. 3. Output of DCT Image Compression.

Original Image (175 KB)

Coefficient=400000(99 KB)

Coefficient=300000(94 KB)

Coefficient=100000(90 KB)

Fig. 4. Output of DCT Compressed Images after different Coefficient Quantization.

Original Image (175 KB)

K=70(92 KB)

K=40(85 KB)

K=20(74 KB)

Fig. 5. SVD Image Compression (Low rank approximation of K values)

Comparison between DCT and SVD Digital Image Compression. The graph below shows a comparison of results between SVD and DCT for the various k values tested:

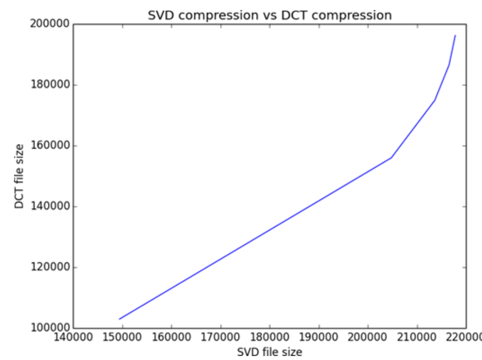

Fig. 6. DCT Compression vs. SVD Compression.

5. Conclusion

Using DCT and SVD techniques, we can save a lot of memory which can be used for other purposes. From our personal computers to the servers of all the big networking websites, all rely heavily on image compression techniques for saving memory. This project firstly dealt with the intricacies and the methodology used in JPEG compression. Then we applied the processes involved in JPEG compression to identify the duplication of images using "discrete cosine transform (DCT)". This project has also applied technique of linear algebra "singular value decomposition (SVD)" to digital image processing. This Project is comparing how the decrease in image quality manifested itself between the two image compression techniques. There is a clear difference in how the low quality SVD compressed images are distorted versus how the low quality DCT compressed images were distorted. SVD has the advantage of providing a good compression ratio, and that can be well adapted to the statistical variation of the image; but it has the disadvantage that it is not fast from the computational point of view. The SVD approach is robust, simple, easy and fast to implement. It works well in a constrained environment. This project is successfully implemented the DCT and SVD for image compression. The system is designed by using MATLAB software (i.e. MATLAB R2018a). This project has been tested for all possible situations on MATLAB environment on Windows 10 and finally produced an 8x8 Compressed DCT image.

References

[1] Anjali Kapoor and Renu Dhir, "Image Compression Using Fast 2-D DCT Technique", *International Journal on Computer Science and Engineering* (IJCSE), vol. 3 pp. 2415-2419, June 2011.

[2] R. C. Gonzalez, R. E. Woods, "Digital Image Processing" Pearson, 3rd Edition, India, 2009.

[3] A.M. Raid, W.M. Khedr, M.A. El-dosuky and wesan Ahmed " JPEG image compression using Discrete Cosine Transform – A survey", International *Journal of Computer Science & Engineering Survey* (IJCSES), vol. 5, no. 2, pp. 29-47, April 2014.

[4] Maneesha Gupta and Amit Kumar Garg, "Analysis Of Image Compression Algorithm Using DCT" *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, pp. 515-521, Jan-Feb 2012.

[5] ISO/IEC 15444-1:2000(E), "Information technology-JPEG 2000 image coding system-Part 1: Core coding system", 2000.

[6] R. C. Gonzalea and R. E. Woods, "Digital Image Processing", 2nd Ed., Prentice Hall, 2004. Singular Value Decomposition Applied To Digital Image Processing- Lijie Cao Division of Computing Studies Arizona State University Polytechnic Campus Mesa, Arizona 85212.

[7] Mande M., Singular Value Decomposition, Department of computer science of the Technological Institute of Bombay India, August 2003.

[8] A.K. Jain," Fundamentals of Digital Image Processing"(1989).

[9] James W. Demmel "Applied Numerical Linear Algebra "(1997).

[10] Karuna Sandhu and Maninder Singh image compression using singular value decomposition (SVD).

[11] H R Swathi, Shah Sohini, Surbhi1 and G Gopichand Image compression using singular value decomposition.

[12] H.S. Prasantha; H. L. Shashidhara; K. N. Balasubramanya Murthy Image Compression Using SVD.

[13] Ali S. S. 2015, A Novel Image Quality Assessment Metric using Singular Value Decomposition, 170–17.

[14] Aishwarya K. M, Ramesh R, Sobarad P M and Singh V 2016 Lossy Image Compression using SVD Coding Algorithm 1384–1389.

[15] Tyler Genter – "Using JPEG to Compress Still Pictures" (2010):

[16] https://mse.redwoods.edu/darnold/math45/laproj/fall2010/tge nter/paperPDFScreen.pdf

[17] http://en.wikipedia.org/Image_compression.

[18] Wei-Yi Wei, "An Introduction to Image Compression."

[19] Subhasis Saha, "Image Compression - from DCT to Wavelets : A Review" http://www.acm.org/crossroads/xrds6- 3/sahaimgcoding.html.

[20] David S. Watkins. "Fundamentals of Matrix Computations" (1991).