



CLASS : B.E. E &TC

SUBJECT: DIVP

EXPT. NO. : 6

DATE: 09/10/2020

TITLE : TO PERFORM IMAGE COMPRESSION USING DCT/DWT

CO 1:	Apply the fundamentals of digital image processing to perform various operations on an image-enhancement in spatial domain/ frequency domain, image-restoration, image compression, video filtering and video compression on a given gray image. Examine the effect of varying the mask size and density of noise in an image and comment on the obtained results.
CO4:	Carry out experiments as an individual and in a team, comprehend and write a laboratory record and draw conclusions at a technical level.

AIM :

1. To apply Discrete Cosine Transform (DCT) on an image.
2. Reconstruct the image by taking Inverse Discrete Cosine Transform (IDCT).
3. Perform DCT based quantization on given 8X8 Matrix.

SOFTWARE REQUIREMENT: Matlab 7.0 or above

6.1THEORY:

6.1.1 Discrete Cosine Transform

In transform coding, a reversible, linear transform like DFT or the discrete cosine transform (DCT) is used to map an image into a set of transform coefficients, which are then quantized and coded for most natural images, a significant number of coefficient have small magnitudes and can be coarsely quantized with a little image distortion.



DCT is a child of DFT, only cosine terms are taken.

- 1D DCT is given by,

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cdot \cos[(2x+1)u \Pi/2N]$$

- 1D IDCT is given by,

$$f(x) = \alpha(u) \sum_{u=0}^{N-1} C(u) \cdot \cos[(2x+1)u \Pi/2N]$$

$$\begin{aligned} \text{where, } \alpha(u) &= \sqrt{1/N} \text{for } u = 0 \\ &= \sqrt{2/N} \text{for } u = 1, 2, \dots, N-1 \end{aligned}$$

Image is a two dimensional signal $f(x, y)$, hence in order to find DCT of an image DCT must be applied in the 2 directions x and y .

- 2D DCT is given by,

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

- 2D IDCT is given by,

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

where,

$$\begin{aligned} \alpha(u) &= \sqrt{1/N} \text{for } u = 0 \\ &= \sqrt{2/N} \text{for } u = 1, 2, \dots, N-1 \\ \alpha(v) &= \sqrt{1/N} \text{for } v = 0 \\ &= \sqrt{2/N} \text{for } v = 1, 2, \dots, N-1 \end{aligned}$$

Consider an image $f(x, y)$ of size $M \times N$, it is subdivided into sub-images of sub-size $n \times n$ and DCT of these sub-images is found out. The DCT of the sub-image is obtained by using the above formula.

6.1.2 Quantization:



Our 8×8 block of DCT coefficients is now ready for compression by quantization. A remarkable and highly useful feature of the JPEG process is that in every step, varying levels of image compression and quality are obtainable through selection of specific quantization matrices. This enables the user to decide on quality levels ranging from 1 to 100, where 1 gives the poorest image quality and highest compression. As a result, the quality/compression ratio can be tailored to suit different needs. Subjective experiments involving the human visual system have resulted in the JPEG standard quantization matrix. With a quality level of 0, this matrix renders both high compression and excellent decompressed image quality.

6.1.3 Advantages:

- Transform co-efficient are relatively uncorrelated.
- Energy is highly compacted.
- Reasonable robust relative to channel error.
- There is no loss in the data.

6.1.4 Disadvantages:

While the DCT-based image coders perform very well at moderate bit rates, at higher compression ratios, image quality degrades because of the artifacts resulting from the block based DCT scheme.

6.1.5 Applications:

The DCT is used in JPEG image compression, MPEG, DV and Theora video compression. DCT is also widely used in signal and image processing, especially for lossy data compression, because it has a strong energy function.



6.2 Algorithm:

6.2.1 Algorithm for part A:

1. Start.
2. Read an image and divide it in 8x8 blocks.
3. Apply 2D DCT formula on each block, resultant image is DCT of image.
4. Divide DCT image into 8x8 blocks.
5. Apply 2D IDCT formula on each block, resultant image is recovered image.
6. Stop.

6.2.2 Algorithm for part B:

1. Start.
2. Take 8X8 Matrix.
3. Leveled off the matrix by subtracting 128 from each entry. (Matrix M)
4. Obtain DCT of above matrix by using the 2D DCT formula. (Matrix D)
 - a. Obtain the coefficient matrix by using formula

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases} \quad (\text{Matrix T}).$$

- b. Obtain DCT Matrix, $D = TMT'$.

5. Select Quantization level, and initialize standard Q matrix. (Matrix Q)
6. Obtain Quantization by dividing each element in the transformed image matrix D

by the corresponding element in the Q matrix. And rounding off to the nearest integer value. (This is quantized matrix can be further coded).

$$C(i,j) = \text{round}(D(i,j) / Q(i,j));$$

7. For Decompression, $R(i,j) = Q(i,j) \times C(i,j)$
8. Obtain the IDCT of above matrix, and round off it to the nearest integer



value.

9. Add 128 to each element.

10. Compare original and recover image.

(Note: If the $M \times N$ size image has to be compressed, divide the image into 8×8 blocks and repeat the above procedure for each block.)

6.3 Conclusion:

In this experiment I applied DCT to an image and observed its histogram. After performing IDCT, I observed its histogram and noticed that there is no difference between the original image and the image obtained after IDCT. Hence DCT can be used for image compression.

6.4 References:

- i. Gonzalez R, Woods R, "Digital image processing", Pearson Prentice Hall, 2008.
- ii. Gonzalez R, Woods R, Steven E, "Digital Image Processing Using MATLAB®", McGraw Hill Education, 2010.
- iii. Jayaraman S, Esakkirajan S and Veerakumar T, "Digital Image Processing" Tata McGraw Hill, 2010
- iv. Joshi, Madhuri A. "Digital Image Processing: an algorithm approach", PHI Learning Pvt. Ltd., 2006.
- v. Pictures taken from:
http://www.imageprocessingplace.com/root_files_V3/image_databases.html

(Course Teacher)



CLASS	: B.E (E &TC)	COURSE	: DIVP
AY	: 2020-21 (SEM- I)	DATE	: 09/10/2020
EXPT. NO.	: 6	CLASS & ROLL NO	: BE VIII 42410
TITLE	: TO PERFORM IMAGE COMPRESSION USING DCT/DWT		

I. CODE:

```
# -*- coding: utf-8 -*-
import cv2
import numpy as np
from matplotlib import pyplot as plt
def main():
    # Reading the input image
    path="C:/Users/Danish/Desktop/DIVP/Images/tajmahal1.jpg"
    img = cv2.imread(path, 0)
    imf = np.float32(img)/255.0 # float conversion/scale
    dct1 = cv2.dct(imf)         # the dct
    dct = np.uint8(dct1*255.0)  # convert back to int
    idct = cv2.idct(dct1)      # take idct
    error = imf - idct
    idct = np.uint8(idct*255.0) #convert back to int
    error = np.uint8(error*255.0)
    plt.figure(0)
    plt.imshow(img, cmap=plt.cm.gray)
    plt.title('Original Image')
    plt.figure(1)
    plt.imshow(dct, cmap=plt.cm.gray)
    plt.title('DCT of Image')
    plt.figure(2)
    plt.hist(img.ravel() , bins=256)
    plt.title("Histogram of an original image")
    plt.figure(3)
    plt.hist(dct.ravel() , bins=256)
    plt.title("Histogram of DCT ")
    plt.figure(4)
    plt.imshow(idct , cmap=plt.cm.gray)
    plt.title("IDCT")
    plt.figure(5)
    plt.hist(idct.ravel() , bins=256)
    plt.title("Histogram of IDCT ")
    plt.figure(6)
    plt.imshow(error , cmap=plt.cm.gray)
    plt.title("Error between original and after taking IDCT")
main()
```

II. RESULTS:

