LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No. 10**

**A.1 Aim:**

Write a program to apply various DFT and DCT transforms on an image and compare the results.

**A.2 Prerequisite:**

1 Matlab programming syntax (Refer the Matlab manual).

2. Knowledge Fast Fourier Transform, DFT and DCT Transforms.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Understand the fundamentals of DFT and DCT Transforms and its effects on digital images.
2. Apply and verify the correctness of DFT and DCT Transforms on images.
3. Identify applications of transforms studied.

**A.4 Theory:**

**A.4.1. Discrete Fourier Transform**

In spatial domain, we perform convolution of filter mask with image data. In frequency domain we perform multiplication of Fourier transform of image data with filter transfer function.

The general idea is that the image (***f(x,y)*** of size ***M***x***N***) will be represented in the frequency domain (***F(u,v)***). The equation for the two-dimensional discrete Fourier transform (DFT) is:

DFT equation

The concept behind the Fourier transform is that any waveform can be constructed using a sum of sine and cosine waves of different frequencies. The exponential in the above formula can be expanded into sines and cosines with the variables ***u*** and ***v*** determining these frequencies.

The inverse of the above discrete Fourier transform is given by the following equation:

Inverst DFT equation

Thus, if we have ***F(u,v)***, we can obtain the corresponding image (***f(x,y)***) using the inverse, discrete Fourier transform.

Things to note about the discrete Fourier transform are the following:

* the value of the transform at the origin of the frequency domain, at ***F(0,0)***, is called the dc component
  + ***F(0,0)*** is equal to ***MN*** times the average value of ***f(x,y)***
  + in MATLAB, ***F(0,0)*** is actually ***F(1,1)*** because array indices in MATLAB start at 1 rather than 0
* the values of the Fourier transform are complex, meaning they have real and imaginary parts. The imaginary parts are represented by i, which is defined solely by the property that its square is −1, ie:http://www.cs.uregina.ca/Links/class-info/425/Lab5/Equations/imaginary_definition.png
* we visually analyze a Fourier transform by computing a **Fourier spectrum** (the magnitude of ***F(u,v)***) and display it as an image.
  + the Fourier spectrum is symmetric about the origin
* the fast Fourier transform (FFT) is a fast algorithm for computing the discrete Fourier transform.
* MATLAB has three functions to compute the DFT:
  + fft -for one dimension (useful for audio)
  + fft2 -for two dimensions (useful for images)
  + fftn -for n dimensions
* MATLAB has three related functions that compute the inverse DFT:
  + idft
  + idft2

**A.4.2. Discrete Cosine Transform (DCT)**

The DCT is similar to the discrete Fourier transform. It transforms a signal or image from the spatial domain to the frequency domain.

The NxN cosine transform matrix C={c(k,n)},also called as discrete cosine transform (DCT), is defined as

Properties of the cosine transform

1. The cosine transform is real and orthogonal, that is

C=C\* where C\* is Complex Conjugate of C.

1. C-1 = CT orthogonality property.
2. The cosine transform is a fast transform as it is real.
3. The cosine transform has excellent energy compaction for highly correlated data.

**A.5 Procedure/Algorithm:**

**A.5.1:**

**TASK 1:**

1. Read the i/p image

2. Resize the image to convert it into square matrix

3. Transform the image using DFT.

4. Verify and note the matrix content of the transformed image in workspace.

5. Display the transformed image.

6. Display the magnitude and phase images out of the transformed image

7. Label each output appropriately.

8. Regenerate and display the original image back

9. Compare the input and output images w.r.t. its matrix content and visibility

on the screen

10. Add the original image with the transformed output image (before applying

inverse transform on it) separately (for all 3 outputs) and observe the result.

11. Save and close the file and name it as **EX9\_Task2\_your Roll no.m**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No.: N230 | Name: Rishul Ghosh |
| Class : MBA Tech CS 3rd year Div. B | Batch : A |
| Date of Experiment: 6-10-21 | Date of Submission: 19-10-21 |
| Grade : | Time of Submission: |
| Date of Grading: |  |

**B.1 Software Code written by student:**

img=double(imread('seeds.jpg'));

[r,c] = size(img);

Output=zeros(r,c);

SumInner = 0;

SumOutner = 0;

%%2D Discrete Fourier Transform

for u = 1:(r-1)

for v = 1:(c-1)

for x = 1:(r-1)

for y = 1:(c-1)

SumInner = SumInner + img(x,y) \* exp(-1i\*2\*3.1416\*((u\*x/r)+(v\*y/c)));

end

SumOutner = SumOutner + SumInner;

end

Output(u,v) = SumOutner;

SumOutner = 0;

SumInner = 0;

end

end

%%Calculate Spectrum and show

Output2 = zeros(r,c);

for u = 1:(r-1)

for v = 1:(c-1)

Output2(u,v) = sqrt((real(Output(u,v))^2+imag(Output(u,v))^2))/1000000;

end

end

output3=dct2(img);

subplot(1,3,1);imshow(img);title('Original Image');

subplot(1,3,2);imshow(Output2);title('DFT');

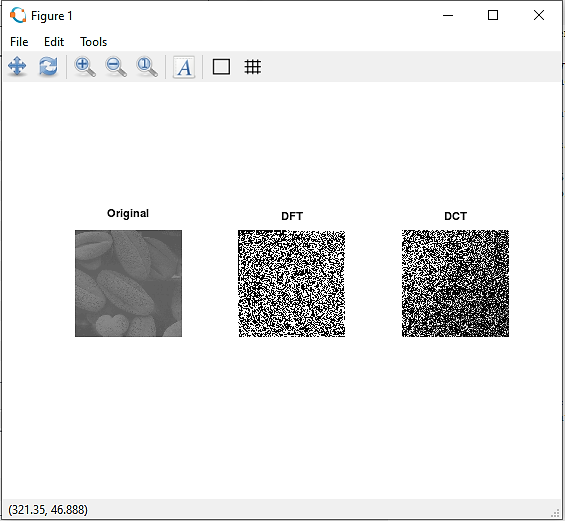
subplot(1,3,3);imshow(Output3);title('DCT');

**B.2 Input and Output:**

**Input Images:**

****

**Output Images:**



**B.3 Observations and learning:**

From the above experiment, we observed and learned to implement transformation into digital image, using DCT and DFT.

**B.4 Conclusion:**

Thus, the aim of implementation of image transformation techniques is completed.

**B.5 Question of Curiosity**

**Q1: What are the applications [apart from given in Q2.] of each of these transforms you have studied?**

Applications of DFT

* Spectrum Analysis of a Sinusoid: Windowing, Zero-Padding, and FFT.
* Spectrograms.
* Filters and Convolution.
* Correlation Analysis.
* Power Spectral Density Estimation.

Applications of DCT

The applications are encoding, decoding, video, audio, multiplexing, control signals, signaling, and analog-to-digital conversion.

**Q2: How can DCT be used for data compression?**

The DCT works by separating images into parts of differing frequencies. During a step called quantization, where part of compression actually occurs, the less important frequencies are discarded, hence the use of the term “lossy.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*