**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech/BTI/MBA**

**Sem: VII ACAY: 2020-21**

**EXPERIMENT NO. 9**

**Aim:**

To write a program in PYTHON to implement DCT on an image

**Software:**  PYTHON.

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Transforms-DCT |

**Outcome:**

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

1. Understand the significance of transforms in image processing
2. Implement forward and inverse DCT on a 2D matrix

**Theory:**

If is the transformation matrix and is its transpose, forward transform of any 2D matrix is given by

Similarly, inverse transform is given by

**Discrete Cosine Transform**

The N X N Cosine Transform Matrix is given by the following expression

|  |
| --- |
| Name of the Experiment: To implement DCT on an image |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance: 11/09/2020 Date of Submission: 11/09/2020 |

**CODE:**

import numpy as np

from skimage import io

import matplotlib.pyplot as plt

from scipy.fftpack import dct,idct

from skimage.color import rgb2gray

def dct2(a):

return dct(dct(a,axis=0, norm='ortho'),axis=1,norm='ortho')

def idct2(a):

return idct(idct(a,axis=0, norm='ortho'),axis=1,norm='ortho')

image\_ori=io.imread('cat.png')

image=rgb2gray(image\_ori)

image=image\*255

io.imshow (image,cmap ='gray')

sh = image.shape

row = sh[0]

col = sh[1]

image\_dct = np.zeros ((row,col),dtype=int)

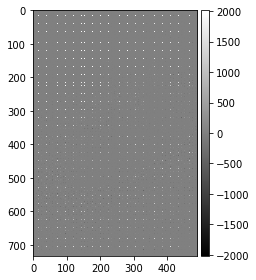
for r in range (0,row//8):

for c in range (0,col//8):

image\_dct[r\*8:(r+1)\*8, c\*8:(c+1)\*8]=dct2(image[r\*8:(r+1)\*8, c\*8:(c+1)\*8])

io.imshow (image\_dct, cmap = 'gray')

image\_dct



th=100

image\_comp = np.zeros ((row,col),dtype=int)

for r in range (row):

for c in range (col):

if image\_dct [r][c]<th:

image\_comp[r][c]=0

else:

image\_comp[r][c]=image\_dct[r][c]

image\_comp

image\_inv = np.zeros ((row,col),dtype=int)

for r in range (0,row//8):

for c in range (0,col//8):

image\_inv[r\*8:(r+1)\*8, c\*8:(c+1)\*8]=idct2(image\_comp[r\*8:(r+1)\*8, c\*8:(c+1)\*8])

io.imshow (image\_inv,cmap='gray')



nz= np.sum ([image\_comp!=0])

print (nz)

CR = (row\*col)/nz

print (CR)



CONCLUSIONS:

1. If all DCT coefficients are retained, then IDCT of the image and the oringinal image is same.

2. To achieve image compression, some of the DC coefficients can be reduced to 0. Such coefficients are not stored to save storage capacity.

3. If threshold is increased, then the quality of image detoriates and compression ratio increases.