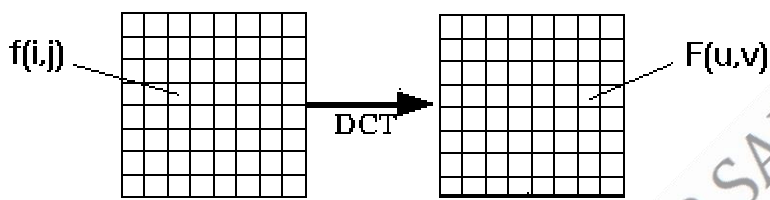
Image Transform (DCT)

Aim:- To perform Discrete Cosine transform on a small matrix of an image.

Theory:-

The Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain (Fig 7.8).



DCT Encoding

The general equation for a 1D (N data items) DCT is defined by the following equation:

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \Lambda(i).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] f(i)$$

and the corresponding *inverse* 1D DCT transform is simple $F^{-1}(u)$, i.e.:

where

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

The general equation for a 2D (N by M image) DCT is defined by the following equation:

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i).\Lambda(j).cos\left[\frac{\pi.u}{2.N}(2i+1)\right] cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$

and the corresponding *inverse* 2D DCT transform is simple $F^{-1}(u,v)$, i.e.:

where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

The basic operation of the DCT is as follows:

- The input image is N by M;
- f(i,j) is the intensity of the pixel in row i and column j;
- F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.
- For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.
- Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.
- The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level;
- 8 bit pixels have levels from 0 to 255.
- Therefore an 8 point DCT would be:

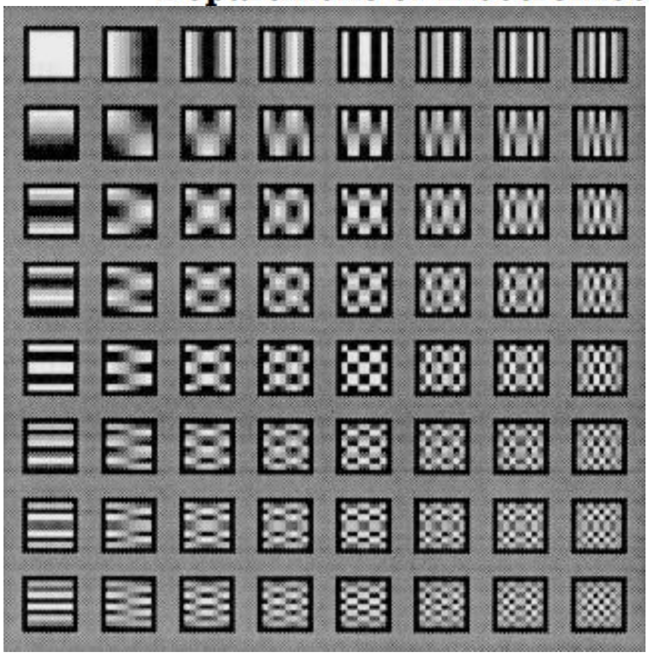
  where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

  Question: What is F[0,0]?

  answer: They define DC and AC components.

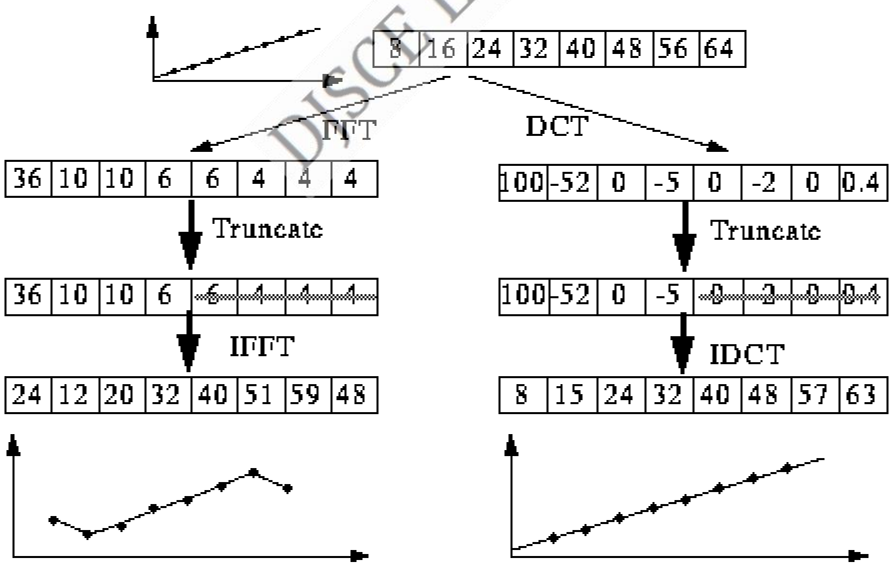
- The output array of DCT coefficients contains integers; these can range from -1024 to 1023.
- It is computationally easier to implement and more efficient to regard the DCT as a set of **basis functions** which given a known input array size (8 x 8) can be precomputed and stored. This involves simply computing values for a convolution mask (8 x8 window) that get applied (summ values x pixelthe window overlap with image apply window accros all rows/columns of image). The values as simply calculated from the DCT formula. The 64 (8 x 8) DCT basis functions are illustrated in Fig 7.9.



DCT basis functions

- Why DCT not FFT?

  DCT is similar to the Fast Fourier Transform (FFT), but can approximate lines well with fewer coefficients (Fig 7.10)

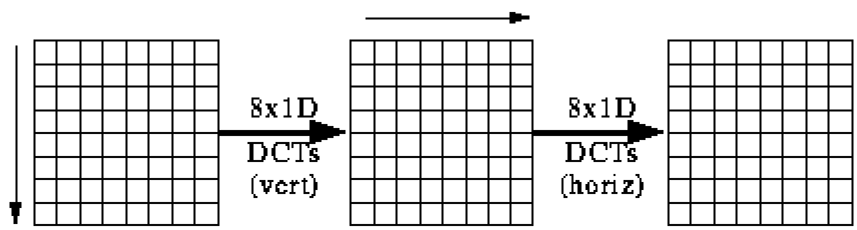


DCT/FFT Comparison

- Computing the 2D DCT
  - Factoring reduces problem to a series of 1D DCTs (Fig 7.11):

- apply 1D DCT (Vertically) to Columns
- apply 1D DCT (Horizontally) to resultant Vertical DCT above.
- or alternatively Horizontal to Vertical.

The equations are given by:



Implementation Instructions:-

Consider a small matrix representing a small part of image & perform 2 dimensional DCT operation on it & observe the result. Do not use direct matlab/python function.

```
%Experiment 8 - Image Transform (DCT)
%Krisha Lakhani - 60001200097

clc;
clear all;
% Original  Matrix
a = [
    0 1 2 1;
    1 2 3 2;
    2 3 4 3;
    1 2 3 2
];

t = zeros(4, 4);
N = 4;

% Calculating DCT Transformation Matrix
for u = 1:N
    for v = 1:N
        if u == 1
            t(u,v) = 1/sqrt(N);
        else
            t(u,v) = (sqrt(2/N)) * cos((((2*(v-1))+1)*pi*(u-1))/(2*N));
        end
    end
end

% Calculating DCT Matrix
dct = t*a*transpose(t);

disp("Krisha Lakhani - 60001200097")
disp("Original matrix:");
disp(a);
disp("DCT transformation matrix(T):");
disp(t);
disp("DCT matrix:");
disp(dct);
```

Command Window

```
Krisha Lakhani - 60001200097
Original matrix:
     0     1     2     1
     1     2     3     2
     2     3     4     3
     1     2     3     2

DCT transformation matrix(T):
    0.5000    0.5000    0.5000    0.5000
    0.6533    0.2706   -0.2706   -0.6533
    0.5000   -0.5000   -0.5000    0.5000
    0.2706   -0.6533    0.6533   -0.2706

DCT matrix:
    8.0000   -1.8478   -2.0000    0.7654
   -1.8478         0    0.0000    0.0000
   -2.0000         0    0.0000    0.0000
    0.7654    0.0000    0.0000   -0.0000
```

Conclusion:

Successful implementation of Discrete Cosine Transform on a 4x4 image (which is a part of an image) has been carried out and DCT Matrix has been achieved with the help of transformation matrix.