Discrete Cosine Transform

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). it transforms an image from the spatial domain to the frequency domain. It widely uses in image compression

Given an image,S, in the spatial domain, the pixel at coordinates (x,y) is denoted S_{xy} .To transform S into an image in the frequency domain,F, we can use the following

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{else} \end{cases}$$

$$C_v = \text{(similar to the above)}$$

$$F_{vu} = \frac{1}{4} C_v C_u \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} S_{yx} cos\left(v\pi \frac{2y+1}{2N}\right) cos\left(u\pi \frac{2x+1}{2N}\right)$$

Example blocks in spatial and frequency domains The example computes the two-dimensional DCT of 8-by-8 blocks in an input image, discards (sets to zero) all but 10 of the 64 DCT coefficients in each block, and then reconstructs the image using the two-dimensional inverse DCT of each block. The example uses the transform matrix computation method

After getting the image compute the two -dimensional DCT of 8 by 8 blocks in the image.the function dctmtx returns the N by N DCT transform matrix

```
T = dctmtx(8);

dct = @(block_struct) T * block_struct.data * T';
B = blockproc(I,[8 8],dct);
```

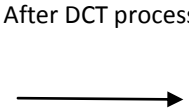
Discard all but 10 of the 64 DCT coefficients in each block

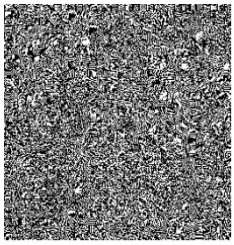
```
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);
```

Reconstruct the image using the two dimensional inverse DCT of each block.

```
invdct = @(block_struct) T' * block_struct.data * T;
I2 = blockproc(B2,[8 8],invdct);
```

After DCT process





Fast Fourier transforms (FFT)

FFT is applied to convert an image from the image (spatial) domain to frequency domain. The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The equation of the FFT to (n x m) matrix.

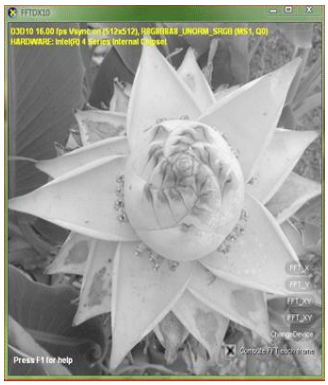
$$F(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(x\frac{m}{M}+y\frac{n}{N})}$$

$$f(m,n) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F(x,y)e^{j2\pi(x\frac{m}{M}+y\frac{n}{N})}$$

Computing the FFT of images using code

```
I=imread('imagename.pbm');
F=fft2(double(I));
S=fftshift(F);
L-log2(S);
A=abs(L);
imagesc(A)
```

Before FFT



After FFT