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.

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^{-}

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

$^{1}(u,v)$, i.e.: where

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.

Properties of the Cosine Transform

1. The cosine transform is real and orthogonal, that is, $C = C^* \Rightarrow C^{-1} = C^T$

2. The cosine transform is not the real part of the unitary DFT. This can be seen by inspection of C and the DFT matrix F. (Also see Problem) However, the cosine transform of a sequence is related to the DFT of its symmetric extension (see Problem)

3. The cosine transform is a fast transform. The cosine transform of a vector of N elements can be calculated in O ($N \log_2 N$) operations via an N-point FFT [19]. To show this we define a new sequence u (n) by reordering the even and odd elements of u (n) as

4. The basis vectors of the cosine transform (that is, rows of C) are the eigenvectors of the symmetric tridiagonal matrix Q_c, defined as

$$Q_c = \begin{pmatrix} 1-\alpha & -\alpha & 0 \\ -\alpha & 1 & 1-\alpha \\ 0 & \alpha & 1-\alpha \end{pmatrix}$$

5. The N X N cosine transform is very close to the KL transform of a first-order stationary Markov sequence of length N whose covariance matrix is given by when the correlation parameter ρ is close to 1. The reason is that R^{-1} is a symmetric tridiagonal matrix, which for a scalar

$$\begin{pmatrix} 1-\rho\alpha & -\alpha & 0 \end{pmatrix}$$

$$\beta^2 R^{-1} = \begin{pmatrix} -\alpha & 1 & 1-\alpha \\ 0 & \alpha & 1-\rho\alpha \end{pmatrix}$$

This gives the approximation $\beta^2 R^{-1} \cong Q_C$ for $\rho \cong 1$

Therefore, if we calculate the N-point inverse FFT of the sequence □(k) □(k) exp (j□k/2N), we can also obtain the inverse DCT in O (N log N) operations. Direct algorithms that do not require FFT as an intermediate step, so that complex arithmetic is avoided, are also possible [18]. The computational complexity of the direct as well as the FFT based methods is about the same.

6. The cosine transform has excellent energy compaction for highly correlated data..

The basis vectors of the cosine transform (that is, rows of C) are the eigenvectors of the sy Therefore, if we calculate the N-point inverse FFT of the sequence □(k) □(k) exp (j□k/2N), we can also obtain the inverse DCT in O (N log N) operations. Direct algorithms that do not require FFT as an intermediate step, so that complex arithmetic is avoided, are also possible [18]. The computational complexity of the direct as well as the FFT based methods is about the same.

Hence the eigenvectors of R and the eigenvectors of Q_c, that is, the cosine transform, will be quite close. These aspects are considered in greater depth in Section on sinusoidal transforms.

This property of the cosine transform together with the fact that it is a fast transform has made it a useful substitute for the KL transform of highly correlated first order Markov sequences.

The two dimensional Discrete Fourier Transform:

Discrete Fourier transform (DFT) is the array given by

$$F(u,v) = \frac{1}{MN} \sum^{M-1} \sum^{N-1} f(x,y) e^{-j 2\pi(ux/M + vy/N)}$$

$x=0 \quad y=0$

$u = 0, \square, M-1, \; v = 0, \square, N-1$

A function represented in either of these forms and can be completely reconstructed via an inverse process with no loss of information.

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\exp(-j2\pi ux)dx \qquad j = \sqrt{-1}$$

1-D Fourier Transformation and its Inverse

If there is a single variable, continuous function f(x) , then Fourier transformation F (u) may be given as and the reverse process to recover f(x) from F(u) is

$$F(u) = R(u) + jI(u) \qquad F(u) = |F(U)|e^{j\phi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad \text{or} \quad \phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

F(u) in polar coordinates

Fourier Transformation and its Inverse . The Fourier Transform of a two dimensional continuous function f(x,y) (an image) of size M * N is given by

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

Inverse Fourier transformation is given by equation

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty}\int F(u,v)\exp[j2\pi(ux+vy)]dudv$$

Where (u,v) are frequency variables.

Preprocessing is done to shift the origin of F(u,v) to frequency coordinate (m/2,n/2)

which is the center of the M*N area occupied by the 2D-FT. It is known as frequency rectangle.

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u)\exp[\, j2\pi ux\,]du$$

Fourier transformation of a discrete function of one variable f(x), x=0, 1, 2, m-1 is given by

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\exp\left[-j2\pi ux/N\right] \text{ for } u=0,1,2,\ldots..,N-1$$

to obtain f(x) from F(u)

$$f(x) = \sum_{u=0}^{N-1} F(u)\exp\left[j2\pi ux/N\right] \quad \text{for } x=0,1,2,\ldots\ldots.,N-1$$

The above two equation (e) and (f) comprise of a discrete Fourier transformation pair. According to Euler's formula

. $e^{jx} = \cos x + j \sin x$

Substituting these value to equation (e)

$F(u)=\Sigma f(x)[\cos 2\pi ux/N+j\text{Sin } 2\pi ux/N]$ for

$u=0,1,2,\ldots\ldots,N-1$

Now each of the m terms of F(u) is called a frequency component of transformation

"The Fourier transformations eparates a function into various components, based on

frequency components. These components are complex quantities.

$$F(u) = R(u) + jI(u) \qquad F(u) = |F(U)|e^{j\phi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad \text{or} \quad \phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

F(u) in polar coordinates

The Fourier Transform of a two dimensional continuous function f(x,y) (an image) of size M * N is given by

$$\mathcal{F}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

Inverse Fourier transformation is given by equation

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty}\int F(u,v)\exp[j2\pi(ux+vy)]dudv$$

Where (u,v) are frequency variables.

Preprocessing is done to shift the origin of F(u,v) to frequency coordinate (m/2,n/2) which is the center of the M*N area occupied by the 2D-FT. It is known as frequency rectangle.