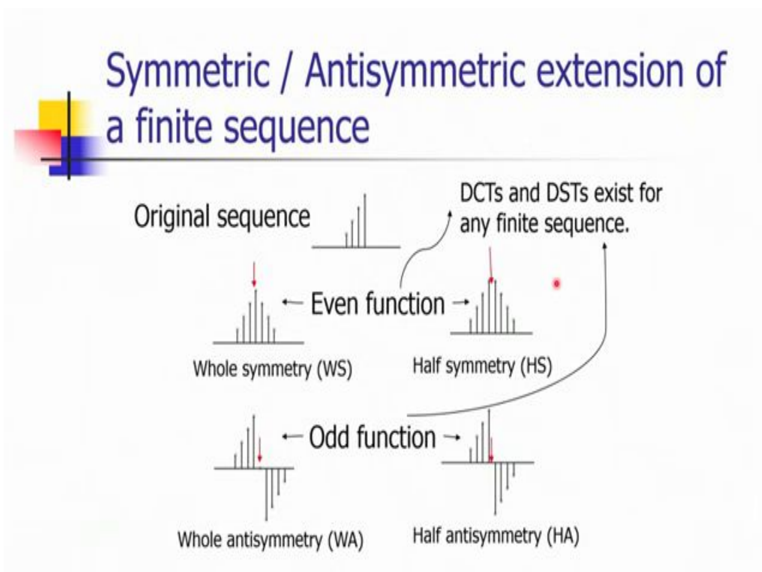
Computer Vision
Prof. Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 04
Image Transform - Part – II

So, let us discuss about the second part of the Image Transform properties and in the previous lecture I discussed how discrete Fourier transform can be generalized. And then we will see using those generalized discrete Fourier transform, we can derive also discrete cosine and discrete sine transforms for discrete sequences and they form a complete base; so you can get a completely construction of those sequences.

(Refer Slide Time: 00:54)



So, to understand that fact let us consider a concept of symmetric or anti-symmetric extension of a finite sequence. As I mentioned for the case of discrete Fourier transform also that given a finite sequence; you can define this sequence in the zone where in the zone of undefined region as per your convenience, make that sequence having certain properties which will be useful and which will be convenient for performing transformation on the sequences; on those extend sequences.

So, one such extension type of extension that is used in that is used for this purpose is symmetric and antisymmetric extension. So, let us understand what this is. you know symmetric extension and what is antisymmetric extension. Now you can see this is an example of a symmetric extension, as this was my original sequence and here the center of symmetry lies at the end sample of the original sequence itself.

And we call this kind of symmetric extension is whole symmetric extension or whole symmetry; a given a sample 4 actually can see that we require another additional 3 samples and by maintaining the symmetric symmetry of the sample values; we can create the whole symmetry. In the other kind of; no symmetry we can create that is called half symmetry, in this case central lies separated with an interval of half of the sampling period or sampling interval from the end sample; this is the point there is center of symmetry.

So, at this towards its left you have 4 samples towards its right also we have 4 samples; so this is called half symmetry. And for the antisymmetry we also similarly defined whole antisymmetry; in whole antisymmetry the antisymmetry, the center of antisymmetry lies at the at a sample value which is introduced; so this value should be 0.

And then next of the values should be you know determined by the corresponding sequence original sequence; they should be negative of the corresponding sample value and this is a whole antisymmetry. So, you can see the total length of in this case in this definition; it is 4 plus this is a value introduced 5 and then this is 4; so this is 9; this is a whole antisymmetry an example of whole antisymmetry.

Similarly, we can have half antisymmetry; so instead of introducing a 0 explicitly here; we can assume that center of antisymmetry lies in between half of the interval between these two samples. So, you do not require any additional introduction of 0; only the samples are inverted here and this is called half antisymmetry extension.

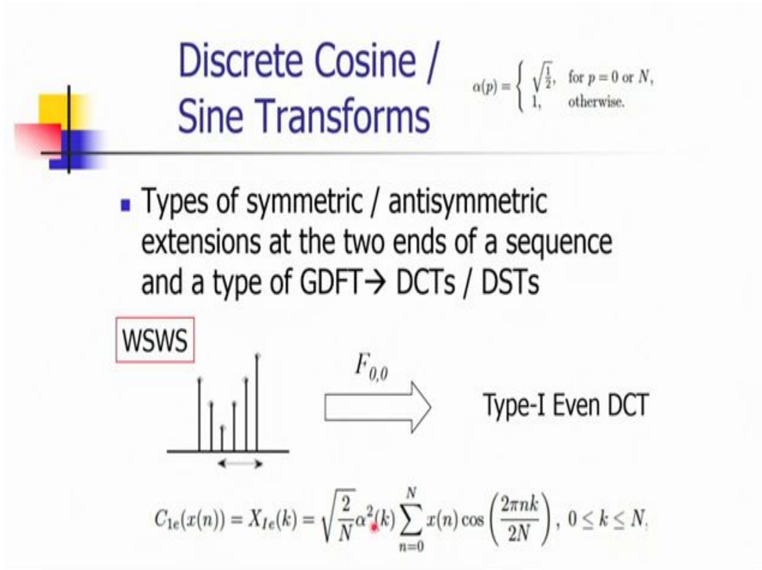
Now, the significance of the this symmetric extension is that, if I make the symmetric extension, then you can observe of this function becomes even function if we consider this is the sample value at the 0th index or x equal to 0 and it becomes also even function if your origin of x lies here. And similarly here it would be odd function once again if your origin lies here and otherwise in this case origin. So, now, as I was mentioning that given an original sequence actually, you have converted this sequence as even function and further if you consider periodic extension of this function; then you can apply discrete Fourier transform on this; some discrete Fourier transform.

And then what happens that since it is an even function that transformation will require only cosine parts; so, you can have discrete cosine transform using this kind of functions here. And again you can reconstruct back using that discrete cosine transform and keep your observation window only on this interval. So, you get the original sequence back.

So, in this way you get all the original sequence by just by using only cosine transforms. Similarly, for odd functions it is sufficient to use only sine functions because for odd functions we have seen that Fourier transform is equivalent to application of only sine functions from the basis function set, here also only the sine functions can be used.

So, that is what using this symmetric antisymmetric extension; we can have DCTs and DSTs or Discrete Cosine Transforms and Discrete Sine Transforms for any finite sequence. And that is the reason why do they exist and you find them in the text books and in many applications. So, for even function it is DCT and for odd function it is DST.

(Refer Slide Time: 06:26)



So, this is what I mean there could be various kinds of discrete cosine transforms because you have so many different varieties. you can have different types of symmetric or antisymmetric extensions at its two ends. And observe whether the sequence become even function or odd function and also you can use different kinds of discrete Fourier transforms from the definition of generalized discrete Fourier transforms.

So, you can have different DCTs and different DSTs; so take this example suppose we have a symmetric extension; whole symmetric extension both at the both ends, in that case we could observe that the functional values would be you know symmetric around this particular value. This is the end and this is the period that would be defined by this symmetric extension; if I extend it periodically with this one, we will get a periodic value.

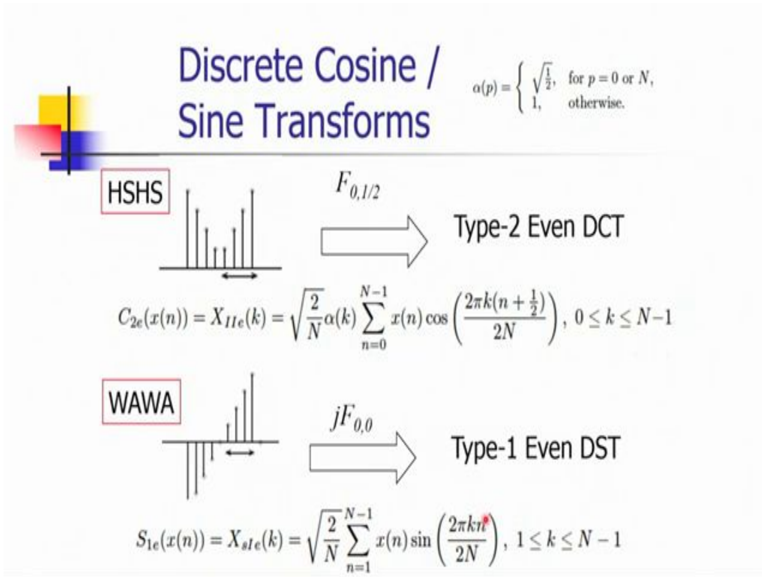
So, this is the original function and using the whole symmetric extension at this end and also at this end; you can identify this is the period, i.e, the minimum period that is formed by this particular extension. And if the value was 4; number of sample was 4 you can see that length of the period becomes 6. So, we will observe that how it affects the corresponding cosine transforms in this case.

It is possible to have a sine transforms and if we apply simply discrete Fourier transform on this extension will get a type I even DCT; whose expression is given in this form. You can say that only the cosine basis function is used; cosine functions is used from the based on the basis vectors. And you can see the period; period is 2N, but you observe the definition of N in this case the value ranges from 0 to capital N which means there are N+1 number of samples. So, if the value of N+1 is equal to 4. So, actual value of N is equal to 3 in this case. So, that is why the period is 6 right;

so this is how the type I even DCT is defined. There is a definition of $\alpha(p)$ also. This is for the normalization operations for making it orthogonal and orthonormal when you are performing the reconstruction; they should satisfy this property, so you will have this particular definition.

$$\alpha(p) = \begin{cases} \sqrt{1/2}, & p = 0 \text{ or } N \\ 0, & \text{otherwise} \end{cases}$$

(Refer Slide Time: 09:15)



So, similarly we can have other kinds of symmetric extensions HSHS. So, at both ends we have half symmetric extension and if this is your original samples that is of length 4; then half symmetric extension that both end will provide you a periodic signal of even function of length 8. And if I apply the alpha that general discrete Fourier transform, when $\alpha=0$ and $\beta = \frac{1}{2}$ that is odd time discrete Fourier transform, then you will find it is a type II even DCT.

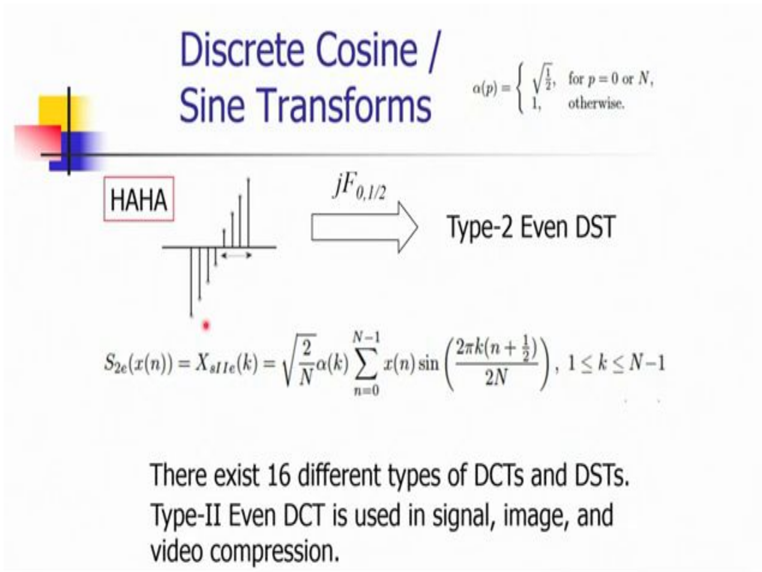
And the expression can be given in this form and here you can observe that given N samples you are generating a period of twice N. So, you have the following expression :

$$C_{2e}(x(n)) = X_{IIe}(k) = \sqrt{\frac{2}{N}}\alpha(k) \sum_{n=0}^{N-1} x(n)\ cos(\frac{2\pi k(n+\frac{1}{2})}{2N}),\ 0 \le k \le N-1$$

and this is a familiar DCT expression what you see in the text book and this is mostly used in image compressions and video compressions. And if I say this is type II even DCT is by default that would be considered as that discrete cosine transforms.

And for discrete sine transform similar we can have antisymmetric extensions; like whole antisymmetry extension and given a 4 samples; you generate a period of 9, it should be (2N + 1). So, we will define the definition of N instead of starting from 1; we should consider 0 we should say N=1, N-1. So, actually your defining it with respect to N-1 samples and that would give you the twice N period and this is what your type I even DST.

(Refer Slide Time: 11:14)

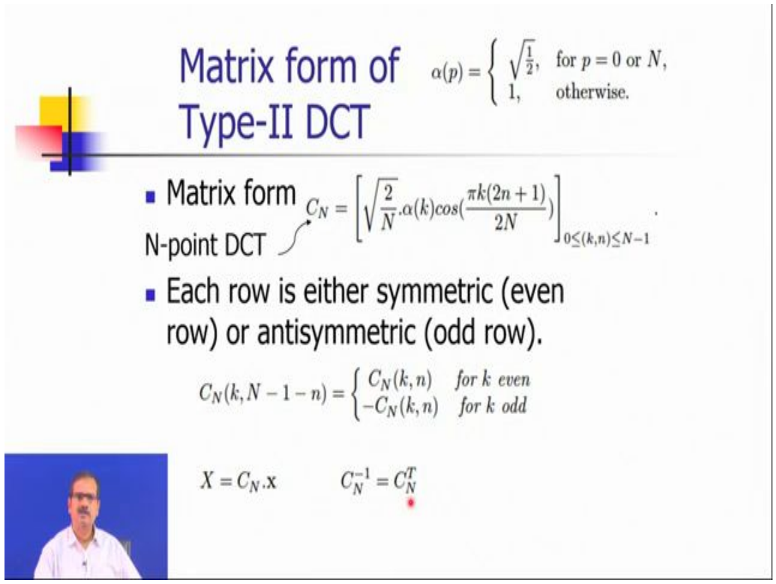


And for Type II even DST similarly, we have half antisymmetric extension and there also we are applying this corresponding discrete Fourier transform and this is a particular transformation matrix that you will be applying there. So, you will get a type II even DST and Following is expression for type II even DST .

$$S_{2e}(x(n)) = X_{sIIe}(k) = \sqrt{\frac{2}{N}}\alpha(k) \sum_{n=0}^{N-1} x(n)\ sin(\frac{2\pi k(n+\frac{1}{2})}{2N}),\ 1 \le k \le N-1$$

There would be many other DCTs and DSTs; in fact, since there are two ends and we can have two varieties of symmetry and two varieties of antisymmetry. So, in total there could be 16 different types of DCTs and DSTs and type II even DCTs is mostly used in signal image and video compression.

(Refer Slide Time: 12:10)



So, in a matrix form we can represent particularly type 2 DCT for an example I am showing all this transformation can be also expressed in the discrete linear transform what we discussed earlier. So, an element of the matrix that is $(k,n)^{th}$ element can be denoted in this form and this matrix is also referred to as N-point DCT matrix which is a type II DCT in this case.

And in this case, there are certain properties which are interesting and which are exploited in various developing different algorithms using DCT coefficients. One of this property is that : each row of this transformation matrix is either symmetric, we call the even row or antisymmetric. And following is a particular equations by which we are expressing this property

$$C_N(k, N-1-n) = C(k,n)\, for\ k\ even$$

$$= -C(k,n)\, for\ k\ odd$$

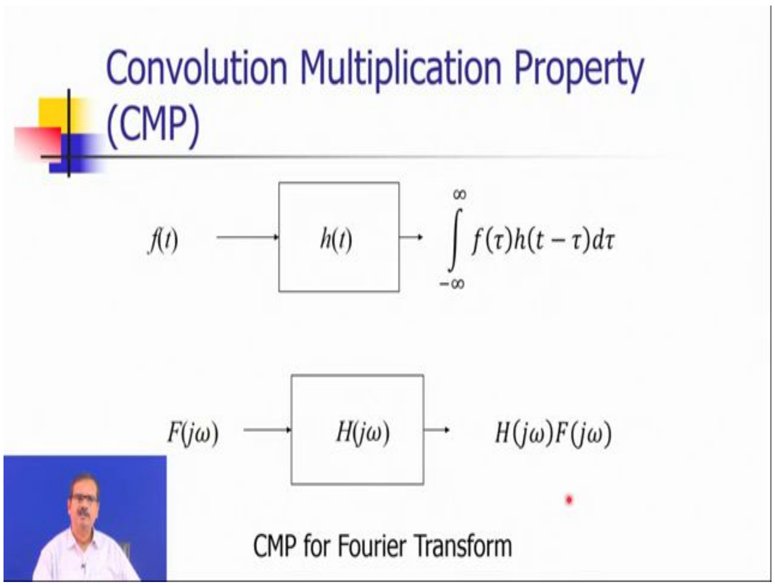
and the transformation can be expressed in terms of multiplication with the column vector $X = C_N x$

and you get the corresponding transformed DCT column vector and its inverse is also given in the following form

$$(C_N)^{-1} = (C_N)^T$$

in this case it is orthonormal expansion; so you can just have the transpose operation.

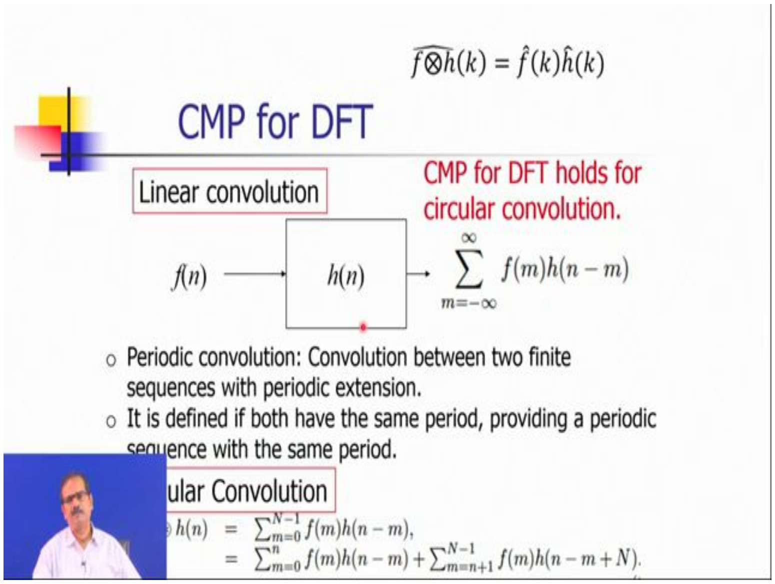
(Refer Slide Time: 13:24)



So, one of the application is that; it can simplify the convolution operations; so what you perform in the functional domain. Because, you can see that convolution operation the functional domain that becomes equivalent to multiplication operation in the transpose domain; when you considered the Fourier transform.

So, this is the particular expressions as these are stage by which we can understand; that the relation is that if I consider a function and if the impulse response of the system is h(t); then convolution of impulse response with that function is equivalent to the product of Fourier transform of this function and Fourier transform of this impulse function. This property is called convolution multiplication property of Fourier transforms.

(Refer Slide Time: 14:19)



Let us consider how this property is reflected using discrete Fourier transform. In the discrete domain the convolution operation; linear convolution operation can be represented in this form instead of integrations now we have summations and these are shifted impulse responses; that is unit impulse responses along the corresponding functional domain at integral points.

And the linear combination of all those shifted values will give you the convolutions as you see that coefficients whose linear combination comes from the functional value itself. So, when you perform the linear convolution; the thing is that, we assume both of the sequence and also the impulse response they are of infinite length. But it happens so, we are handling with finite sequence and discrete Fourier transform is applied for a finite sequence.

So, how do you consider this definitions, how do you we can modify this definition because it is not necessarily the corresponding functional domain where the function is not defined in a finite sequence; it is not necessarily they have to be set to 0. If you set them to 0 it becomes equivalent to linear convolution, but you can consider some periodic extension as we did for discrete Fourier transform definition itself. So, periodic convolution of two finite sequences is defined as convolution between two finite sequences with periodic extension.

Same linear convolution and it can be observed that if they have same period that the periodic sequence also will be of same period. So, with this definition; with this property we can define a circular convolution. So, it is sufficient if we compute the periodic convolutions for a single period only, it need not compute in the whole functional domain itself.

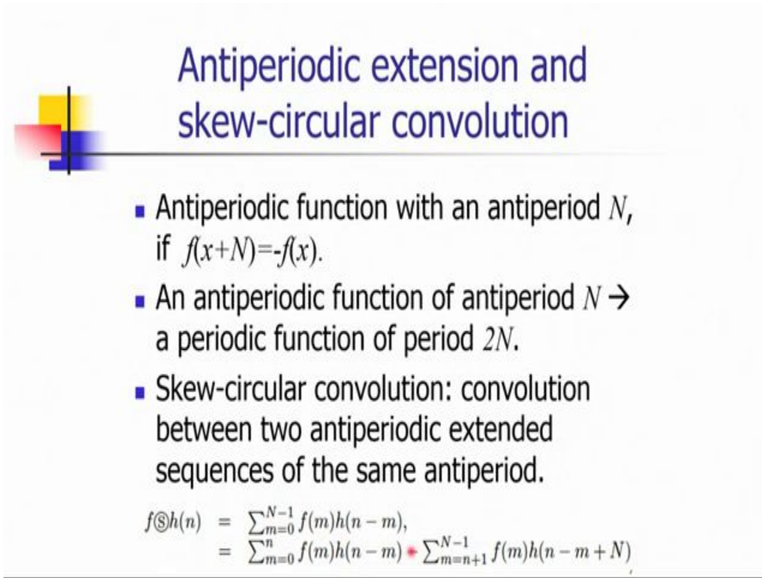
$$f \circledast h = \sum_{m=0}^{N-1} f(m)h(n-m)$$

$$= \sum_{m=0}^{n} f(m)h(n-m) + \sum_{m=n+1}^{N-1} f(m)h(n-m+N)$$

So, in a circular convolution which is as I mentioned, it is a periodic extension and we can compute only with the interval of from 0 to N-1. And if I apply this periodic periodicity definition; then this can be broken into the two parts as shown above. This is the definition of a circular convolution. The interesting part is that convolution multiplication property for DFT holds for the circular convolution.

So if I consider the impulse response of a system and functional finite sequence both should be of same length; take the discrete Fourier transform and take the product point wise at the corresponding coefficient wise take the product. Then, you will get also the transform coefficients of the sequence what would have been the output of the system if they are all periodic extensions.

(Refer Slide Time: 17:36)



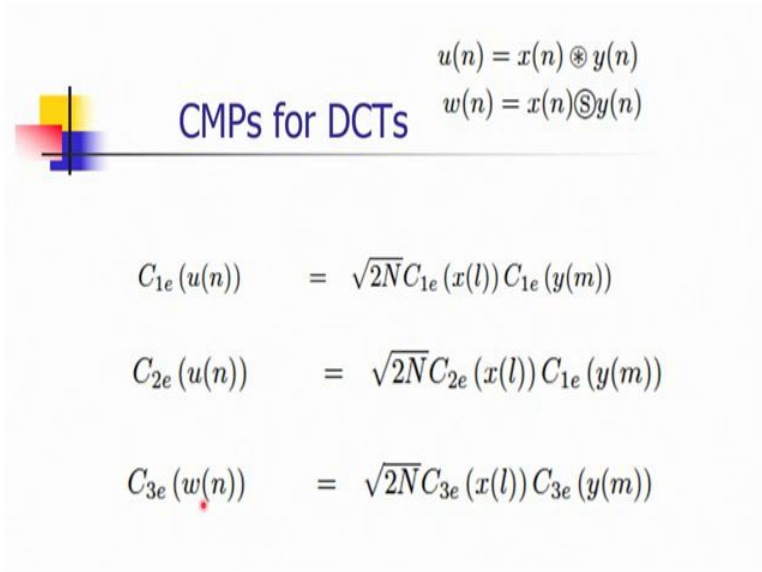
So, this is what is your is the property in the discrete Fourier transform domain. So, what about; there is another kind of convolution with antiperiodic extension. Like periodic extension, we can have antiperiodic extension; in the antiperiodic extension first we perform the this antiperiodic extension over the functional domain that is with an antiperiod N.

$$f(x+N) =- f(x)$$

This is what it is defined antiperiodic function and if I do this antiperiodic extension; it is it is also a periodic function of period N that is interesting.I I antiperiodic function does not mean it is; it is not periodic. It is actually periodic, but the periodicity is now doubled you can check this things; you can make an antiperiodic extension and we will see there is a periodicity of twice the number of samples of the original sequence.

So, skew circular convolution is defined with respect to this periodic values or we can say this antiperiodic extensions. Because we will be again observing the convolved output only in the observation window of the original sequence. So, a skew circular convolution is defined in this form; you take the convolutions again from the linear convolution definition itself, but you apply the properties of antiperiodic extension; then you will get this expression.

(Refer Slide Time: 19:18)

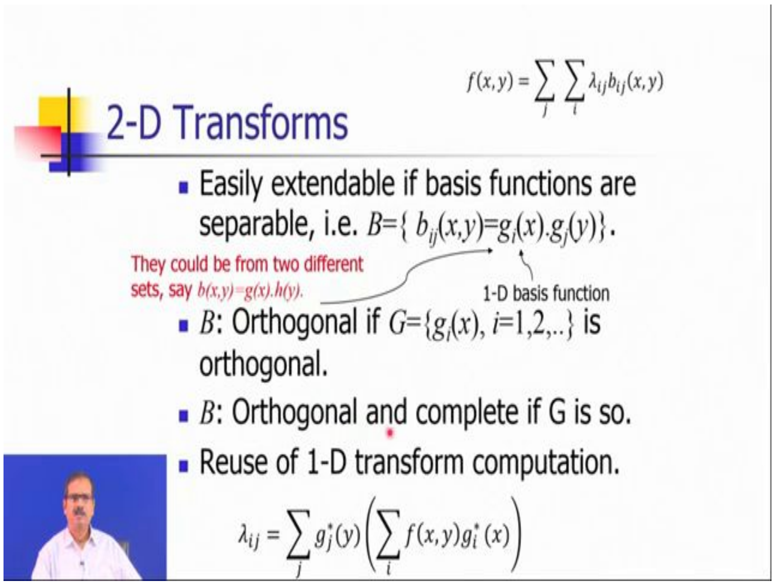


So, using the circular convolution and skew circular convolution; we can find that there are different convolution multiplication properties those hold also in DCTs and DSTs. I will show you some of them for DCTs. For example, you take this case that you have two functions; one of them you can consider as the impulse response say y and both of them of same period; of same length. So, if you take the type I DCT of this one and type I DCT of this one; then if I multiply and no that is that should be multiplied by this factor because of the definition of DCTs what we have in this lecture following that this multiplication factor would there.

And then you get your result in the transform domain itself; that is the type I DCT of the circularly convolved output of these two sequences. Similarly, a type II DCT of this one and type I DCT of impulse response will give you a type II DCT of the corresponding convolved result. You should note that number of samples depend upon the corresponding type of DCT what you are applying. Because finally, there should be of same periodicity; so in this case we will have N+1 samples and there are N samples because N sample define a DCT of 2N period and N+1 samples define it type I DCT of 2N period.

So, that you need to be careful while applying this particular properties and type III DCTs have these interesting property; where actually you can find the output is this, this properties applied for skew circular convolution.

(Refer Slide Time: 21:20)



So, whatever we have discussed in one dimensional transform; this can be easily extended for two dimensional transforms this discussion. If I consider our basis function in two dimension as a certain particular property; which is separability property; so they are separable if I can I write this basis function into this two form; $B = \{b_{ij}(x,y) = g_i(x).g_j(y)\}$

that means, you can write it as a product of 2 basis functions as shown above; they are all independent of independently they can be computed using a particular variable. So, this is an 1 D basis function say product of two one dimensional basis function.

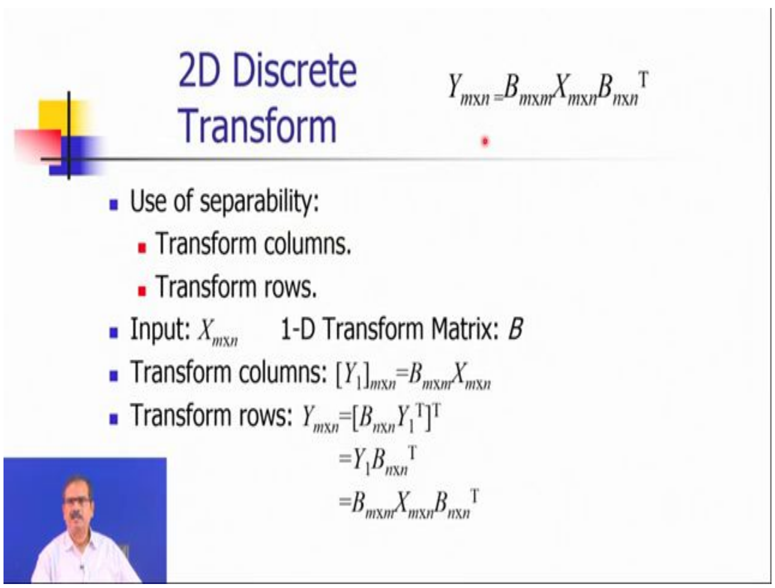
If both of them are orthogonal; then you can see that this set of basis functions will be also orthogonal and then we can reuse this one dimensional transform computation and we can express them in the following equation

$$\lambda_{ij} = \sum_j g_j^*(y)(\sum_i f(x,y) g_i^*(x))$$

So, first we are computing the transforms with respect to the x; by changing the values or sequence with respect to variation over x.

And then we are considering the transform of say with respect to the value of y, we will see that how this computation is reflected in terms of matrix operation. So, you should note here though in this slide; we have used the same notations for this two functions; they could be separate, only thing is that both of them needs to be orthogonal to keep the orthogonality property of this basis functions.

(Refer Slide Time: 23:11)



So, 2D discrete transform can be easily computed by using this separability property. So, let us consider the computation being this way say you can transform columns and then you can transform rows. Suppose, you have a 1 dimensional transform matrix B and there is an input. your input is a corresponding input which is given in terms of $m \times n$ matrix ,that is your data.

So,first I can transform columns of this input data; that is the image in this case $m \times n$, some image block. So, I will be transforming them columns; so, we know that dimension of each column is m So, that is why we are use using the corresponding transformation matrix which deals with m dimensional vectors; which is represented in this form and then each column is now is transformed into another m columns. So, you are doing it for n such columns. So, you will find n dimensional vectors transform columns and there are n columns; so this is a matrix you will get.

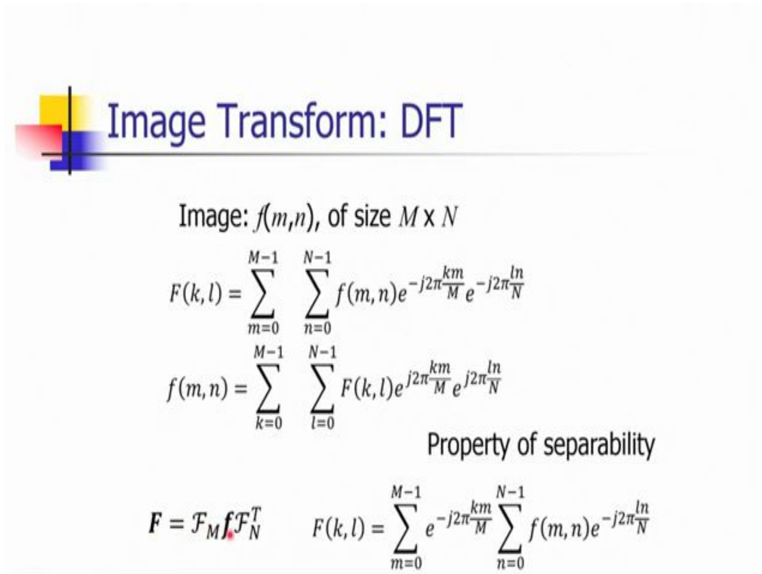
After that what you can do ,you can now transform the rows of this matrix which means now you have to take the transpose and then perform the $n \times n$. Since all rows are n dimensional, so you have to use the n-point transformation matrix here and then you can get the final no transforms of the two dimensional image of $m \times n$ size.

So, if I expand Y_1 I can write the whole operations in the following composite form.

$$Y_{m\times n} = [B_{n\times n}(Y_1)^T]^T = Y_1(B_{n\times n})^T = B_{m\times m}X_{m\times n}(B_{n\times n})^T$$

So, the whole operations can be described in this particular form and that gives me the 2 dimensional discrete form. So, this is how a 2 dimensional discrete form can be represented given the 2 dimensional input image we can use the corresponding transformation matrix towards its right and towards its left in this way and we can get the corresponding transformed image in the transformed domain itself.

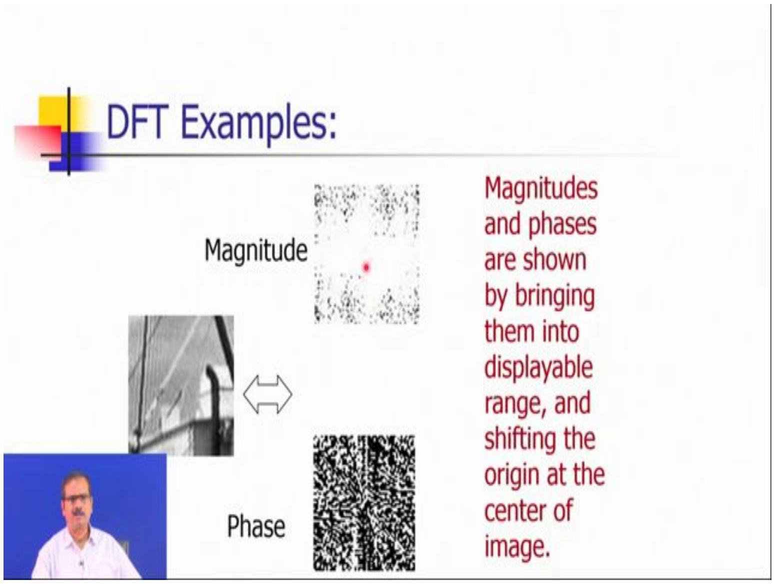
(Refer Slide Time: 25:39)



So, typical examples could be say discrete Fourier transform; we can also express this transforms using the summation operations because they are the basis functions here or the basis vectors here; they are separable, they are expressed in this form. And you can see that it is a simple extension of the discrete Fourier transform of what you had in the in your in the 1 dimensional case. And using property of separability, once again we can simplify this computation and when we express in terms of matrix multiplication; we have already defined the matrix representation of the transformation matrix of DFT.

And we consider this is a $m \times m$ransformation matrix and this is a $n \times n$ matrix; we are just denoting it by simple M showing it as m point discrete Fourier transformation matrix. So, this will give you the corresponding transformation of the image f.

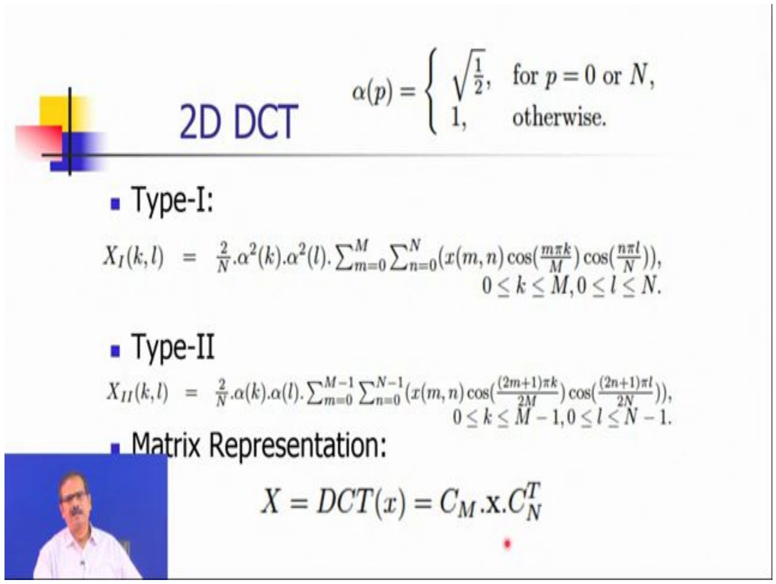
(Refer Slide Time: 26:47)



So, one typical example it is shown here that now given this image; we can perform Fourier transform and it will give me since it is in the complex quantity every transform coefficient in the discrete Fourier transform is a complex element. So, it has its magnitude and phase at every point; so you get two components of this transformation. And in this particular image you should note that if I apply Fourier transform, coefficient values would be very large and it is very difficult to make them display.

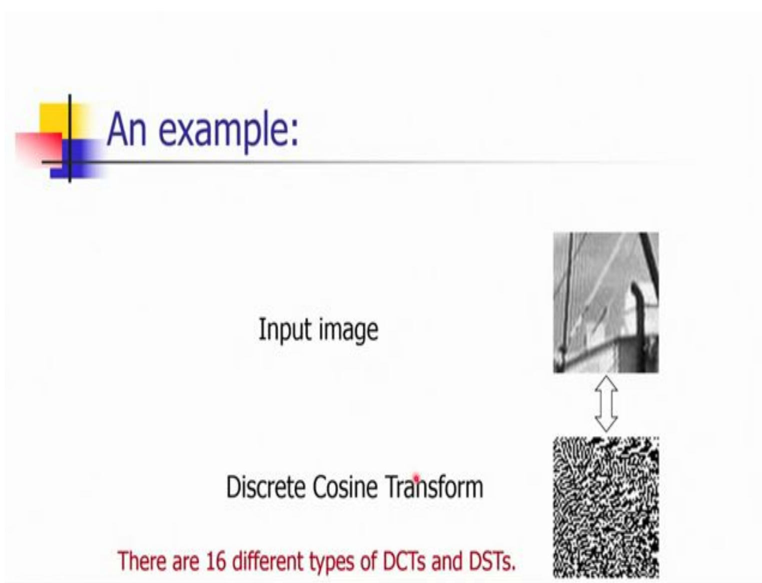
So, we have made them display well and also we have shifted the origin of the transformation space.

(Refer Slide Time: 27:33)



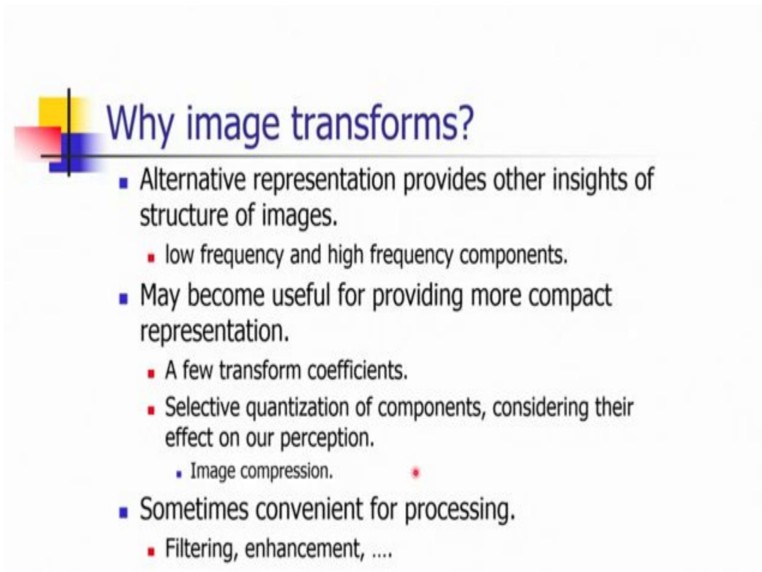
So similar way like discrete Fourier transform, we can define also discrete two dimensional discrete cosine transform. These are same simple extensions what we had in 1 dimension and in the matrix representation; we have the similar representation what we discussed for when we are extending a 1 dimensional transform to a 2 dimensional transform.

(Refer Slide Time: 28:00)



This is an example of a DCT; so if given this input image, you have this discrete cosine transform. In this case also you have scaling of the coefficient values and there are as I have mentioned there could be 16 different types of DCT and DSTs this is just one typical example which is type II even DCT that is shown here.

(Refer Slide Time: 28:27)



So, I will conclude this lecture by mentioning that why do we require image transforms? As you can see image transform, give an alternative representation of image instead of representing the image in the functional domain itself, we are representing it in a different domain; it gives a different insight of structure of images.

And for example, if I consider the frequency domain representations using Fourier transforms, we get low frequency and high frequency components. So, it may become useful for providing more compact representation; you can use only a few transform coefficients to get a very close approximation of the functional representation or you can perform selective quantization of components, considering their effect on our perception.

And those are used in the algorithms for image compressions, even video compressions. And sometimes many processing becomes convenient when we use this transform coefficients; like we have already discussed about filtering, there are operations like enhancement, restoration etc many other operations; where this transforms are useful. So, with this I end my lecture on image transforms.

Thank you very much for your patience and listening to my lecture.