Digital Image Processing

Prof. P. K. Biswas

Department of Electrical & Electronics Communication Engineering

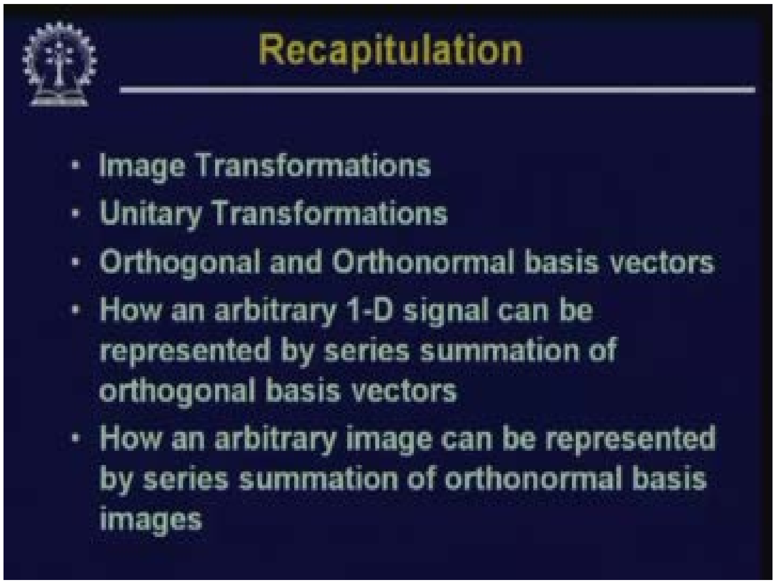
Indian Institute of Technology, Kharagpur

Lecture 12

Image Transformation - II

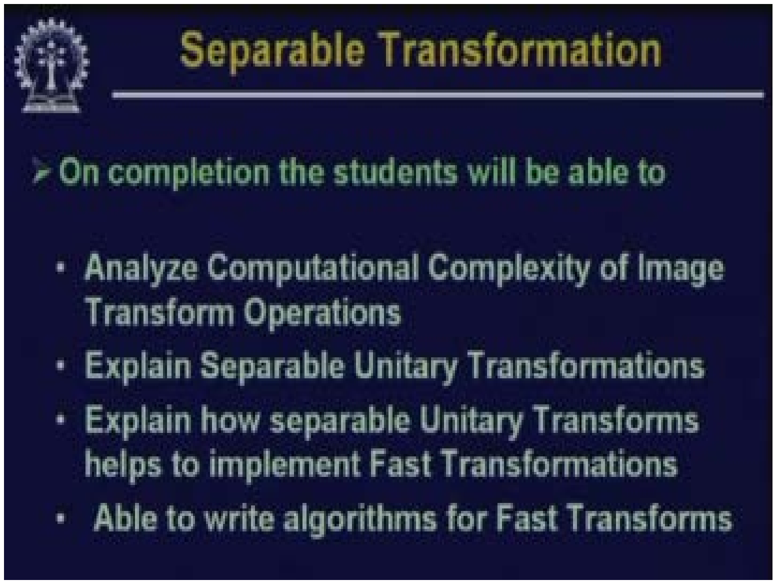
Hello, welcome to the video lecture series on digital image processing. Last class we started our discussion on image transformation. Today we are going to continue with the same topic that is we will continue with the image transformation topic. So, let us see what we have done in our last lecture.

(Refer Slide Time: 1:16)



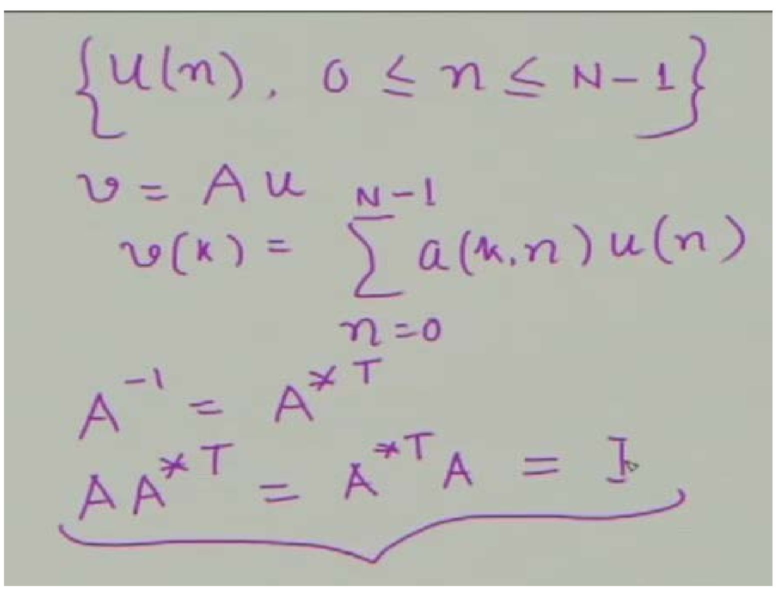
In our introductory lecture on image transformations, we have said the basics of image transformation. We have seen what is meant by a unitary transform. We have also seen what is orthogonal and orthonormal basis vectors. We have seen how an arbitrary 1 dimensional signal can be represented by series of summation of orthogonal basis vectors and we have also seen how an arbitrary image can be represented by series of summation of orthonormal basis images.
So, when we talk about the image transformation; basically, the image is represented as a series summation of orthonormal basis images.

(Refer Slide Time: 2:12)



After today's lecture, the students will be able to analyze the computational complexity of image transform operations. They will be able to explain what is meant by a separable unitary transformation, they will also know how separable unitary transforms help to implement fast transformations and of course, they will be able to write algorithms for fast transforms. So, first let us see that what we have done in the last class.

(Refer Slide Time: 2:51)



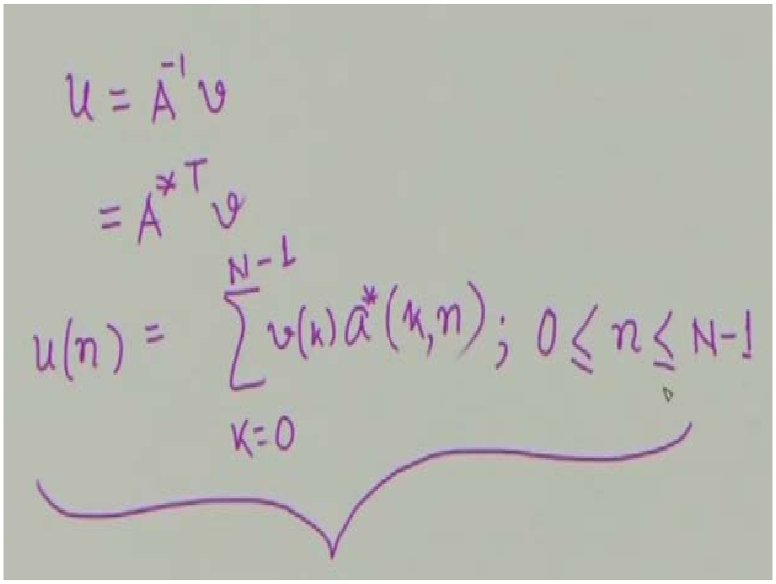
In the last class, we have taken 1 dimensional sequence of the discrete signal samples is given in the form u (n) where n varies from 0 to some capital N minus 1. So, we have taken initially a 1 dimensional sequence of discrete samples like this, that is u (n) and we have found out what is meant by unitary transformation of this 1 dimensional discrete sequence. So, by unitary transformation, by unitary transformation of this 1 dimensional discrete sequence is given by say v is equal to A times u where A is a unitary matrix and this can be represented expanded in the form v (k) is equal to we have a (k, n) u (n) where n varies from 0 to capital N minus 1 assuming that we have capital N number of samples in the input discrete sequence.

Now, we say that this transformation is a unitary transformation if the matrix A is a unitary matrix. So, what is meant by a unitary matrix? The matrix A will be said to be a unitary matrix if it obeys the relation that A inverse, inverse of matrix A will be given by A conjugate transpose. That is if you take the conjugate of every element of matrix A and then the take then take then take the transpose of those conjugate elements; then that should be equal to the inverse of matrix A itself.

So, this says that A into A conjugate transpose, that should be same as A conjugate transpose A which will be same as an identity matrix. So, if this relation is true for the matrix A, then we say that A is a unitary matrix and the transformation which is given by this unitary matrix is unitary transformation. So, using this matrix A, we go for unitary matrix, unitary transformation.

Now, once we have this transformation and we get the transformation coefficients v (k) or the transformed vector, transform sequence v; we should be also able to find out that how from these transformation coefficients, we get back the original sequence u (n). So, this original sequence is obtained by a similar such relation which is given by u is equal to A.

(Refer Slide Time: 6:23)

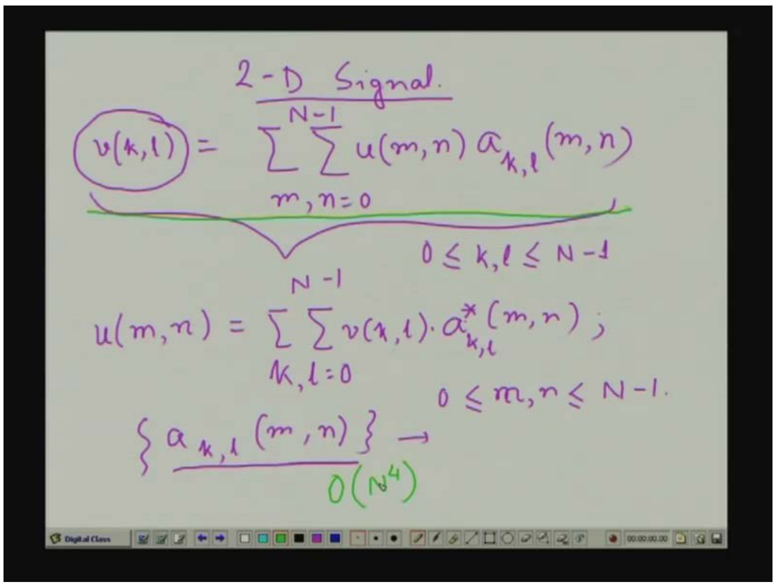


Obviously, it should equal to A inverse v and in our case, since A inverse is same as A conjugate transpose; so, this can be written as A conjugate transpose v and this expression can be expanded as u (n) is equal to summation v (k) a conjugate (k, n) where k varies from 0 to N minus 1 and we have to compute this for all values of n varying from 0 to n minus 1, so 0 less than or equal to n less than or equal to capital N minus 1.

So, by using the unitary transformation, we can get the coefficients, the transformation coefficients and using the inverse transformation, we can obtain the input sequence, input discrete sequence from the coefficient, from this sequence of coefficients. And, this expression says that the input sequence u (n) is now represented in the form of series summation of a set of vectors or orthonormal basis vectors. So, this is what we get in case of 1 dimensional sequence.

Now, let us see what will be the case in case of a 2 dimensional sequence.

(Refer Slide Time: 8:18)



So, for a 2 dimensional sequence; see if I go for the case of 2 dimensional signals; then the same transformation equations will be of the form v (k, l) is equal to we have to have double summation u (m, n) into $a_{k, l}$ (m, n) where both m and n varies from 0 to capital N minus 1.

So here, u (m, n) is the input image, it is a 2 dimensional image. Again, we are transforming this using the unitary matrix A and in the expanded form, the expression can be written like this - v (k, l) is equal to double summation u (m, n) $a_{k, l}$ (m, n) where both m and n varies from 0 to infinity and this has to be computed for all the values of k and l where k and l varies from 0 to n minus 1. So, all k and l will be in the range 0 to N minus 1.

In the same manner, we can have the inverse transformation so that we can get the original 2 dimensional matrix from the transformation coefficient matrix and this inverse transformation in the expanded form can again be written like this. So, from v (k, l) we have to get back u (m, n). So, we can write it as u (m, n) again is equal to double summation v (k, l) into a star $_{k, l}$ (m, n) where both k and l will vary in the range 0 to capital N minus 1 and this we have to compute for all values of m and n in the range 0 to capital N minus 1 where this image transform that is $a_{k, l}$ (m, n), this is nothing but a set of complete orthonormal discrete basis functions. So, this $a_{k, l}$ (m, n), this is a set of complete orthonormal basis functions.

And, in our last class, we have said what is meant by the complete set of orthonormal basis functions and in this case, this quantity the v (k, l), what we are getting these are known as transform coefficients. Now, let us see that what will be the computational complexity of these expressions.

If you take any of these expressions, say for example the forward transformation where we have this particular expression v (k, l) is equal to double summation u (m, n) $a_{k, l}$ (m, n) where m and n vary from 0 to capital N minus 1. That means both m and n; m will vary from 0 to capital N minus 1, n will also vary from 0 to capital N minus 1.

So, to compute this v (k, l), you find that if I compute this particular expression; for every v (k, l), the number of complex multiplication and complex addition that has to be performed is of the order of capital N square and you remember that this has to be computed for every value of k and l where k and l vary in the range 0 to capital n minus 1. That is k is having capital N number of values, l will also have capital N number of values.

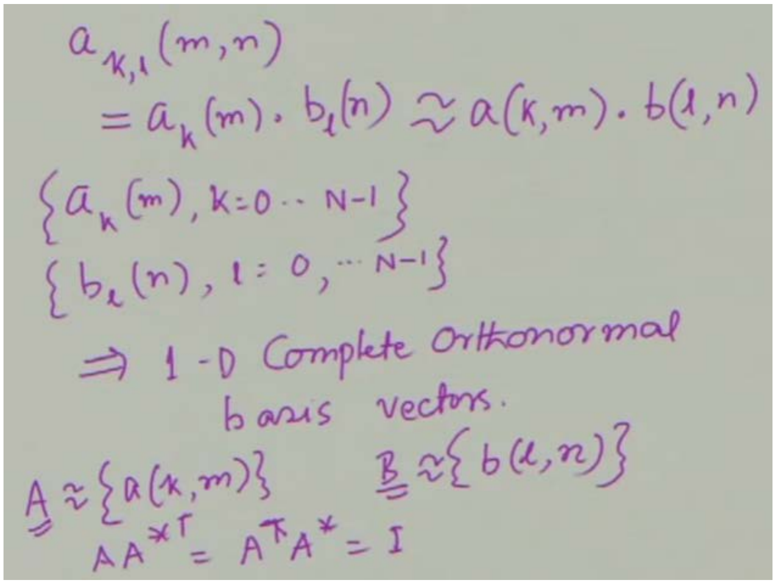
So, to find out v (k, l), a single coefficient v (k, l), we have to have of the order of capital N square number of complex multiplications and additions and because this has to computed for very v (k, l) and we have capital N square number of coefficients because both k and l vary in the range 0 to capital N minus 1; so there are capital N square number of coefficients and for computation of each of the coefficient, we need capital N square number of complex addition and multiplication.

So, the total amount of computation that will be needed in this particular case is of the order of capital N to the power 4. Obviously, this is quite expensive for any of the practical size images because in practical cases, we get images of the size of say 256 by 256 pixels or 512 by 512 pixels, even it can go upto say 1k by 1k number of pixels or 2k by 2k number of pixels and so on.

So, if the computational complexity is of the order of capital N to the power 4 where the image is of size n by n; you find that what is the tremendous amount of computation that has to be performed for doing the image transformations using this simple relation? So, what is the way out? We have to think that how we can reduce the computational complexity?

Obviously, to reduce the computational complexity, we have to use some mathematical tools and that is where we have the concept of separable unitary transforms.

(Refer Slide Time: 15:12)



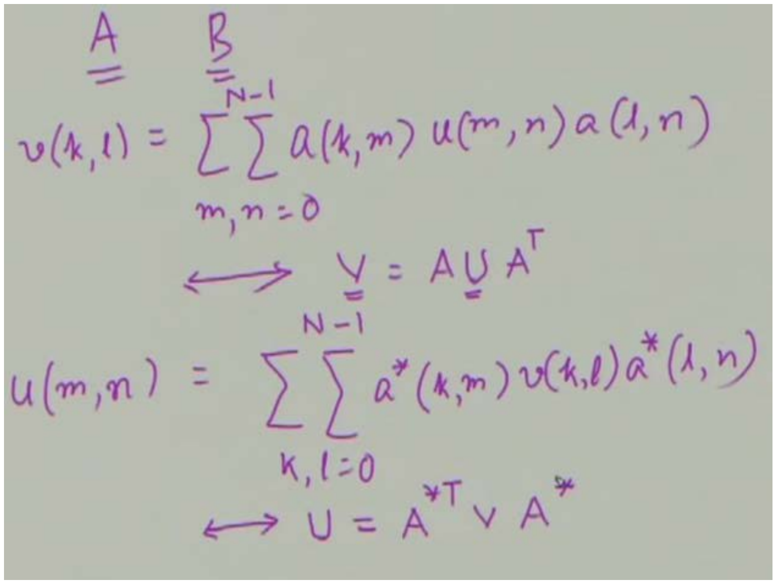
So, we find that we have the transformation matrix which is represented by matrix A or we have represented this as $a_{k,\,l}$ (m, n) and we say that this is separable if $a_{k,\,l}$ (m, n) can be represented in the form, so if I can represent this in the form a_k (m) into say b_l (n) or equivalently, I can put it in the form a (k, m) into b (l, n).

So, if this $a_{k,\,l}$ (m, n) can be represented as a product of a (k, m) and b (l, n); then this is called our then this is called separable. So, in this case, both a (k, m) where k varies from 0 to capital N minus 1 and b (l, n) where l also varies from 0 to capital N minus 1. So, these 2 sets - a (k, m) and b (l, n), they are nothing but 1 dimensional complete orthogonal sets of basis vectors. So, both a (k, m) and b (l, n), they are 1 dimensional complete orthonormal basis vectors.

Now, if I represent this set of orthonormal basis vectors, both a (k, m) and b (l, n) in the form of matrices that is we represent A as a (k, m) as matrix A and similarly b (l, n) the set of these orthonormal basis vectors if we represent in the form of a matrix, then both and both A and B themselves should be unitary matrices and we have said that if they are unitary matrixes, then AA conjugate transpose is equal to A transpose A conjugate which should be equal to identity matrix.

So, if this holds true; in that case, we say that the transformation that we are going to have is a separable transformation and we are going to see next that how this separable transformation helps us to reduce the computational complexity. See, in the original form, we had the computational complexity of the order capital N to the power 4 and will see that whether this computational complexity can be reduced from capital from the order capital N to the power 4.

(Refer Slide Time: 18:42)



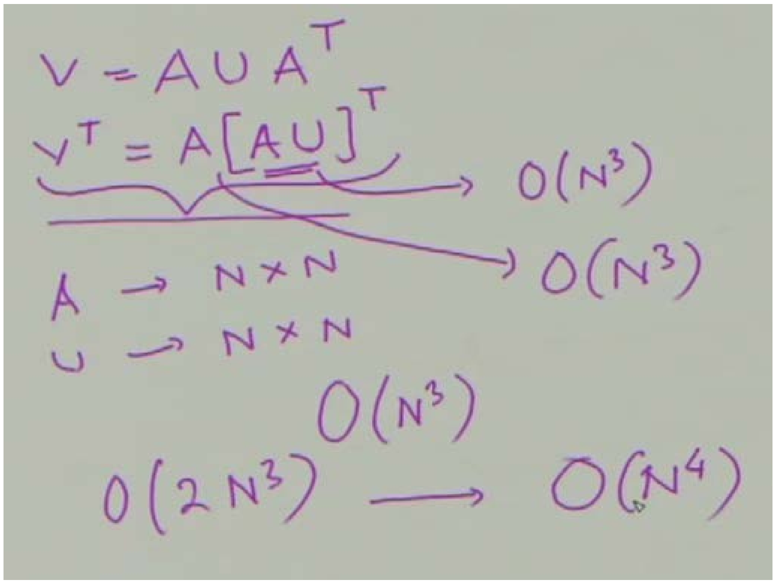
Now, in most of the cases, what we do is we assume these 2 matrixes A and B to be same and that is how these are decided. So, if I take both A and B to be equal to be same, then the transformation equations can be written in the form v (k, l) will be double summation a (k, m) u (m, n) a (l, n).

So, compare this with our earlier expressions where in the expression we had $a_{k,\ l}$ (m, n). So now, this $a_{k,\ l}$ (m, n), we are separating into 2 components. One is a (k, m), the other one is a (l, m) and this is possible because the matrix A that we are considering is a separable matrix. So, because this is a separable matrix, we can write v (k, l) in the form of a (k, m) u (m, n) into a (l, n) where again in this case, both m and n will vary from 0 to capital N minus 1 and in matrix form, this equation can be represented as V equal to AUA transpose where U is the input image of dimension capital N by capital N and V is the coefficient matrix again of dimension capital N by capital N and the matrix A is also of dimension capital N by capital N.

In the same manner, the inverse transformation that is what we have got is the coefficient matrix and by inverse transformation, we want to have the original image matrix from the coefficient matrix. So, in the same manner, the inverse transformation can now be written as u (m, n) equal to again we have to have this double summation a star (k, m) v (k, l) a star (l, n) where both k and l will vary from 0 to capital N minus 1.

So, this is the expression for the inverse transformation and again as before, this inverse transformation can be represented in the form of a matrix equation where the matrix equation will look like this - U equal to A conjugate transpose V into A conjugate and these are called 2 dimensional separable transformations. So, you find that from our original expressions, we have now brought it to an expression in the form of separable transformations.

(Refer Slide Time: 22:13)



So, you find that this particular expression that is V, when we have written this V equal to sorry so here we have written v equal to, so if you go back to our previous slide, you will find that V equal to AU UA transpose. So, if I just write in the form AUA transpose, so I get the coefficient matrix V from our original image matrix U by using this separable transformations. The same equation, we can also represent in the form of V transpose equal to A [AU] transpose.

Now, what does this equation mean? You will find that here what it says that if I compute A, the matrix multiplication of A and U take the transpose of this. Then re multiply that result with the matrix A itself. Then what we are going to get is the transpose of the coefficient matrix V. See, if I analyze this equation, it simply indicates that these 2 dimensional transformations can be performed by first transforming each column U with matrix A and then transforming each row of the result to obtain the rows of the coefficient matrix V. So, that is what is meant by this particular expression.

So, A into U, what it does is it transforms each column of the matrix A with of the input image A with the input image U with the matrix A and this intermediate result you get, you transform each row of this again with matrix A and that gives you the rows of the transformation matrix or the rows of the coefficient matrix V. And, so if I take the transpose of this final result, what we are going to get is the set of coefficient matrix that we wanted to have. Now, if I analyze this particular expression, you will find that A is a matrix of dimension capital N by capital N, U is also a matrix of the same dimension capital N by capital N.

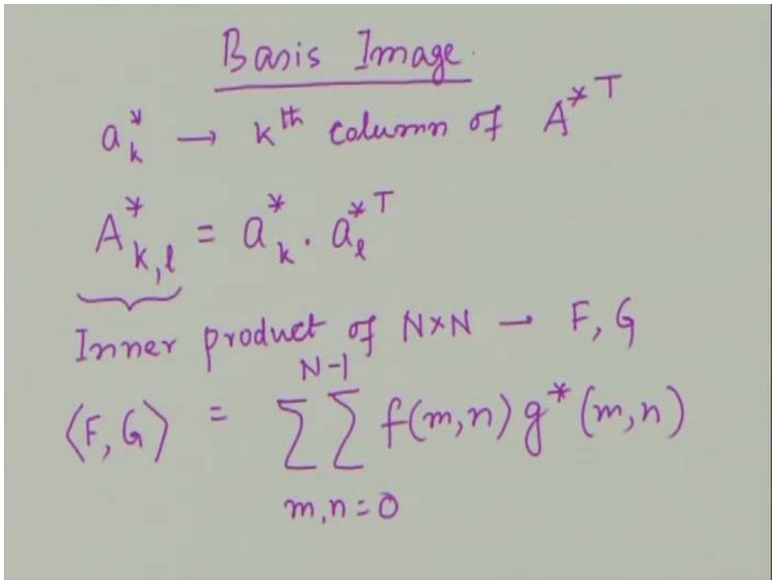
And then, from matrix algebra, we know that if I wanted to multiply 2 matrices of dimension capital N by capital N; then the complexity or the number of additions and multiplications that we have to do is of order capital N cube. So here, to perform this first multiplication, we have to have of order N cube number of multiplications additions. The resultant matrix is also of dimension capital N by capital N and the second matrix multiplication that we want to perform that is A with AU transpose, this will also need of order N cube number of multiplications additions.

So, the total number of addition and multiplication that we have to perform when I implement this as a separable transformation is nothing but of order 2N cube and you compare this with our original configuration when we had seen that the number of addition and multiplication that has to be done is of order N to the power 4. So, what we have obtained in this particular case is the reduction of computational complexity by a factor of capital N.

So, this simply indicates that if the transformation is done in the form of a separable transformation, then it is possible and as we have seen that we can reduce the computational complexity of implementation of the transformation operation. Obviously, the final result that you get that is the coefficient matrix is same as the coefficient matrix that you get when you implement this as a non separable transformation.

So, advantage is that you get by implementing this as a separable transformation is reduction in computational complexity. Now, let us see that what is meant by the basis images.

(Refer Slide Time: 27:36)

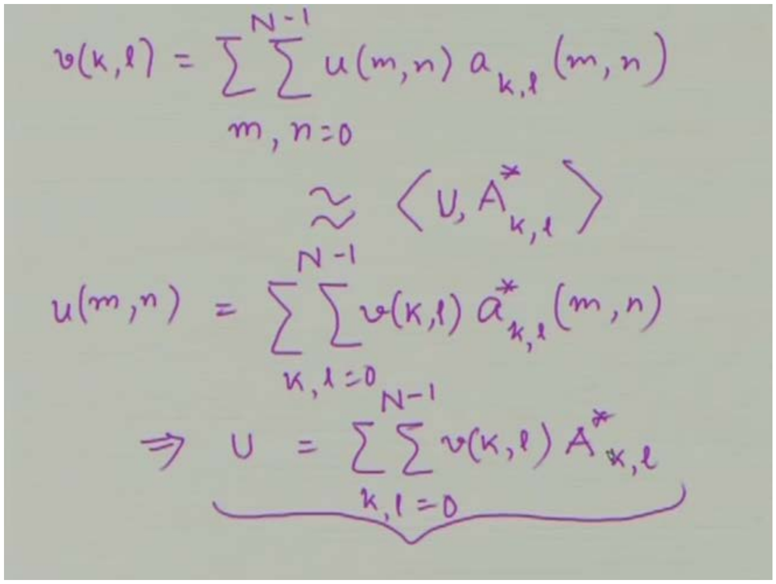


So, what is meant by basis image? Now here, we assume that suppose a_k star, this denote the k’th column of the matrix A conjugate transpose. So, a_k star, we represent this, the k’th column of A conjugate transpose where A is the transformation matrix and now if I define the matrices $A_{k,\,l}$ star as a_k star into a_l star transpose; so you will find that a_k star is the k’th column of the matrix A star transpose, a_l star is also the l’th column of the matrix A conjugate transpose.

So, if I take the product of a_k star and a_l star transpose, then I get the matrix, a matrix $A_{k,\,l}$ star and let us also define the inner product of say 2 N by N matrices. So, I define inner product of 2 N by N matrices, say F and G. So, the inner product of these 2 matrices F and G are defined as f (m, n) g star (m, n) where both m and n vary from 0 to capital N minus 1.

So, define the inner product of 2 matrices F and G in the form of f (m, n) g star (m, n) where both m and n vary from 0 to capital N minus 1.

(Refer Slide Time: 30:21)



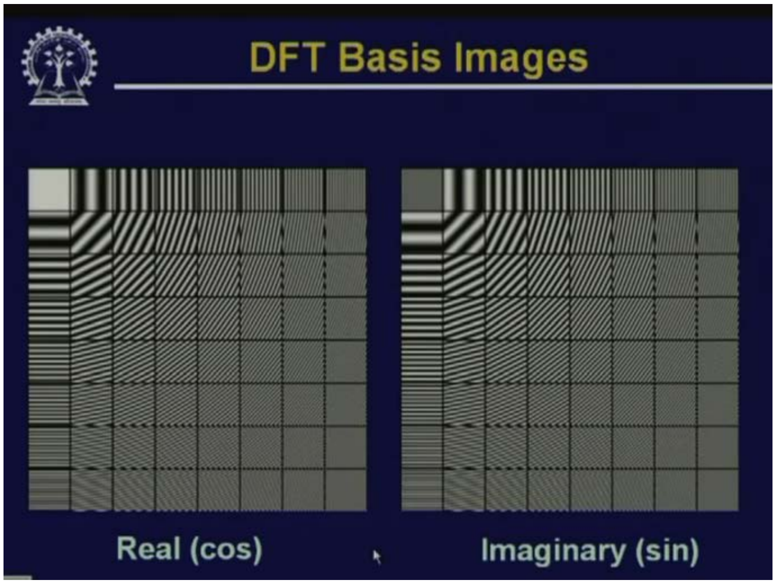
So now, by using these 2 definitions, now if I rewrite all transformation equations; so now we can write the transformation equations as v (k, l) is equal to, you will find that the old expression that we have written u (m, n) $a_{k,l}$ (m, n) where both m and n vary from 0 to capital N minus 1. So, this is nothing but as per our definition, so if you just look at this definition; this is nothing but an expression of an inner product. So, this was the expression of the inner product.

So, this transformation equation is nothing but an expression of an inner product and this inner product is the inner product of the image matrix u with the transformation matrix A star k, l. Similarly, if I write the inverse transformation u (m, n) which is given as again in the form of double summation, v (k, l) into a star k, l (m, n) where k, l vary from 0 to capital N minus 1. So again, you will find that in the matrix form, this will be written as U equal to summation v (k, l) into A star k, l where both k and l vary from 0 to capital N minus 1.

So, if you look at this particular expression, you will find that our original image matrix now is represented by a linear combination of N square matrices A star k, l because both k and l vary from 0 to capital N minus 1. So, I have N square such matrices $A_{k,l}$ and by looking at this expression, you will find that our original image matrix U is now represented by a linear combination of N square matrices a star k, l where each of these N square matrices are of dimension capital N by capital N and these matrices A star k, l are known as the basis images.

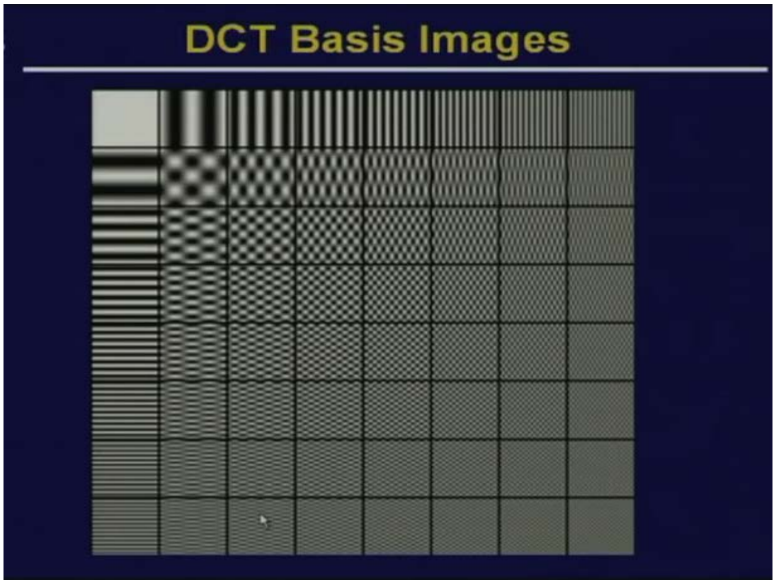
So, this particular derivation simply says that the purpose of image transformation is to represent an input image in the form of linear combination of a set of basis images. Now, to look at how this basis images look like, to see how this basis images look like; let us see some of the images.

(Refer Slide Time: 33:52)



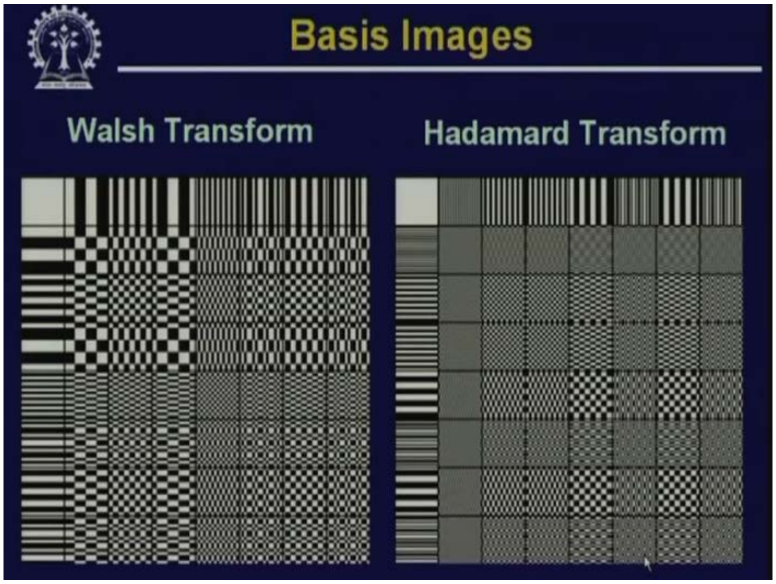
So here, we find that we have shown 2 images. We will see later that these are the basis images of dimension 8 by 8. So here, we have shown basis images of dimension 8 by 8 and there are total 8 into 8 that is 64 basis images. We will see later that in case of discrete Fourier transformation, we get 2 components. One is the real component, other one is the imaginary component. So accordingly, we have to have 2 basis images. One corresponds to the real component, the other one corresponds to the imaginary component.

(Refer Slide Time: 34:35)



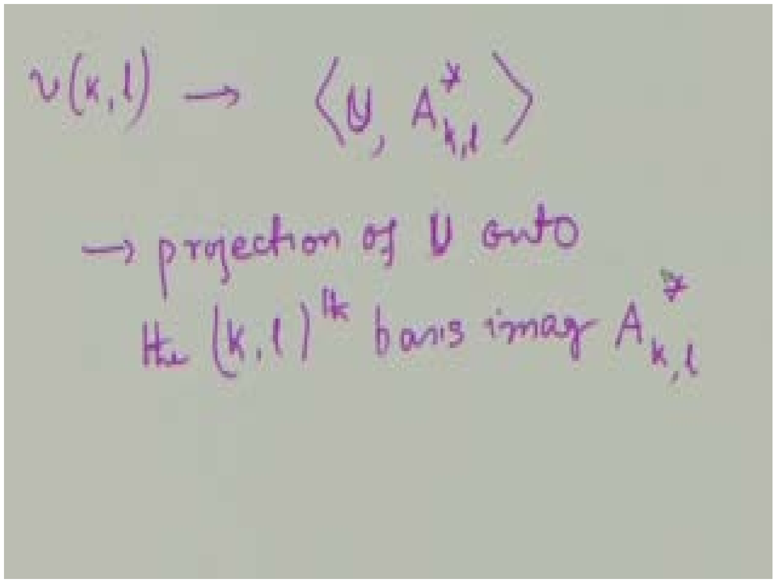
Similarly, this is another basis image which corresponds to the discrete cosine transformation. So again, here I have shown the basis images of size N by 8 by 8. Of course, the inner size image is quite expanded and again we have 8 into 8 that is 64 numbers of images. So here, we find that a row of this represents the index k and the column indicates the index l. So again, we have 64 images, each of these 64 images is of size 8 by 8 pixels.

(Refer Slide Time: 35:22)



Similarly, we have the basis images for other transformations like Walsh transform, Hadamard transform and so on. So, once we look at the basis images; so the purpose of showing these basis images is that as we said that the basic purpose of image transformation is to represent an input image as linear combination of a set of basis images and when we take this linear combination, each of this basis images will be weighted by the corresponding coefficient in the transformation coefficient v (k, l) that we compute after the transformation and as we have said that this v (k, l) is nothing but the inner product of k, l'th basis image.

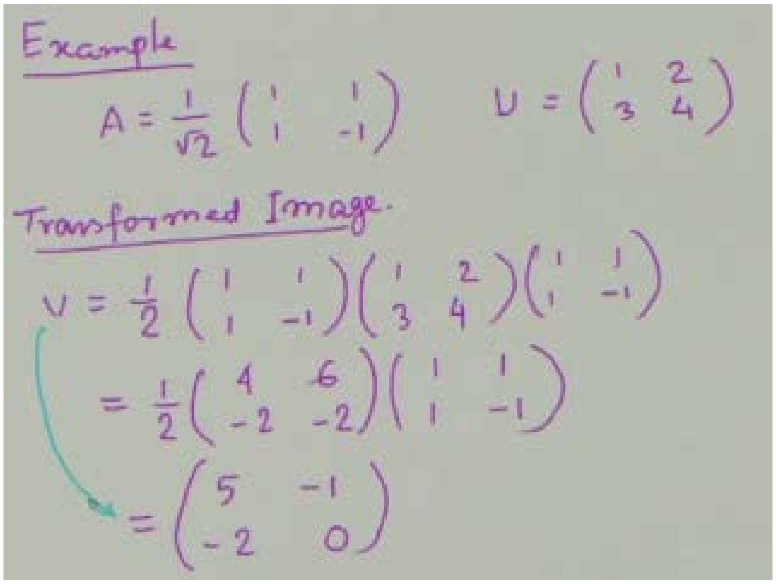
(Refer Slide Time: 36:22)



So, when you compute this v (k, l) as we have seen earlier; so, if you just look at this, this v (k, l) which is represented as inner product of the input image U and the k, l'th basis image is star k, l. So, each of these coefficients v (k, l) is actually represented as the inner product of the input image U with the k, l'th basis image $A_{k,\ l}$ star and because this is the inner product of the input image U and the k, l'th basis image $A_{k,\ l}$ star, this is also called the projection of the input image on the k, l'th basis image.

So, this is also called the projection of the input image U onto the k, l'th basis image $A_{k,\ l}$ star and this also shows that any N by N image; any image input image of size, any input image U of size capital N by capital N can be expanded using a complete set of N square basis images. So, that is the basic purpose of our input of the image transformation. Now, let us take an example.

(Refer Slide Time: 38:24)



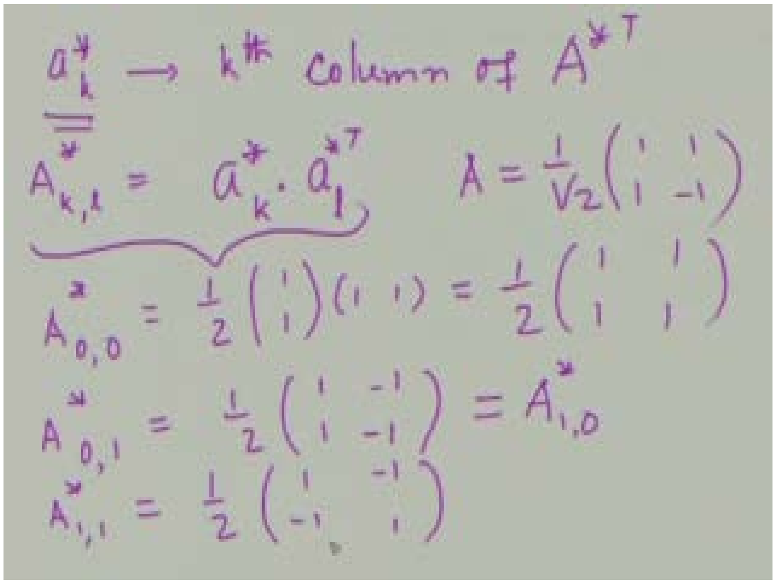
So, let us consider an example of this transformation. Say, we have been given a transformation matrix which is given by A equal to 1 upon root 2 (1, 1, minus 1) and we have the input image matrix U equal to (1, 2, 3, 4) and in this example we will try to see that how this input image U can be transformed with this transform matrix A and the transformation coefficients that you get, If I take the inverse transformation of that, we should be able to get back our original input image U.

So given this, the transformed image; we can compute the transformed image like this, the transformation matrix V will be given by 1 upon 2 into (1, 1, 1, minus 1) into our input image (1, 2, 3, 4). See, if you just see our expressions, you will find that our expression was something like this. When we computed V, we had computed V equal to AUA transpose. So, by using that, we have AU, then A transpose and by nature of this transformation matrix A, you will find that A transpose is nothing but same as A.

So, you will have (1, 1, 1, minus 1) and if you do this matrix computation, it will simply come out to be 1 upon 2 into (4, 6, minus 2, minus 2) into (1, 1, 1, minus 1). And on completion of this matrix multiplication, the final coefficient matrix V will come out to be (5, minus 1, minus 2, 0). So, I get the coefficient matrix V as (5, minus 1, minus 2, 0).

Now, let us see that what is the, for this particular transformation, what will be the corresponding basis images?

(Refer Slide Time: 41:41)

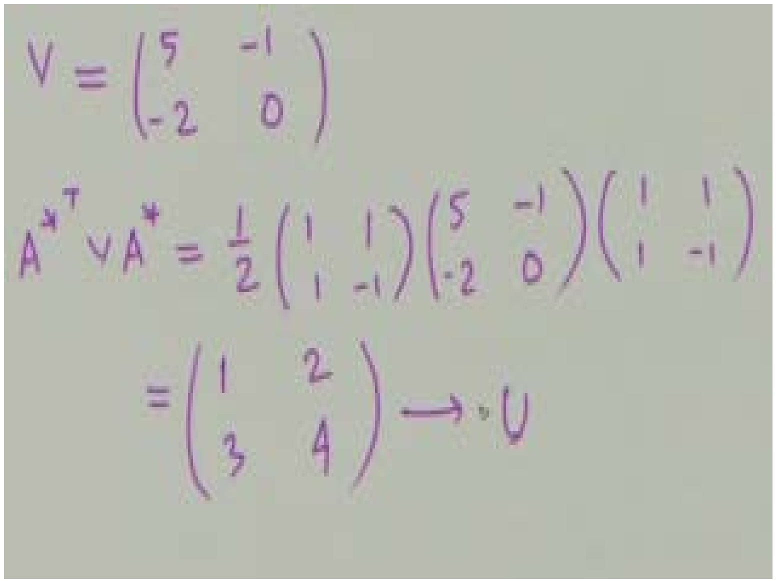


Now, when we defined the basis images, you remember that we have said that we have assumed a_k star to be the k'th column. This was the k'th column of matrix A star transpose. Now, using the same concept and from this our basis functions was taken as $A_{k,\ l}$ star which was given by a_k star multiplied with ah sorry $A_{k,\ l}$ star the k, l'th basis image was computed as a_k star multiplied with a_l star transpose. So, this is how we had computed the basis images, we have defined the basis images.

So, using the same concept, in this particular example where we have all the transformation matrix A is given as 1 upon root 2 (1, 1, 1, minus 1); I can compute the basis images as $A_{0,\ 0}$ star. The 0'Th basis image will be simply half into the basis vectors (1, 1) and (1, 1) transpose. So, this will be nothing but half into (1, 1, 1, 1). Similarly, we can also compute $A_{0,\ 1}$ that is 01'th basis image will be given as half into (1, 1, minus 1, minus 1) which will be same as $A_{1,\ 0}$ that is 10'th basis image and similarly we can also compute $A_{1,\ 1}$ that is 11'th basis image will be come out to be half into (1, minus 1, minus 1, minus 1).

So, this is simply by the matrix multiplication operations. We can compute these basis images from the rows of from the columns of A conjugate transpose.

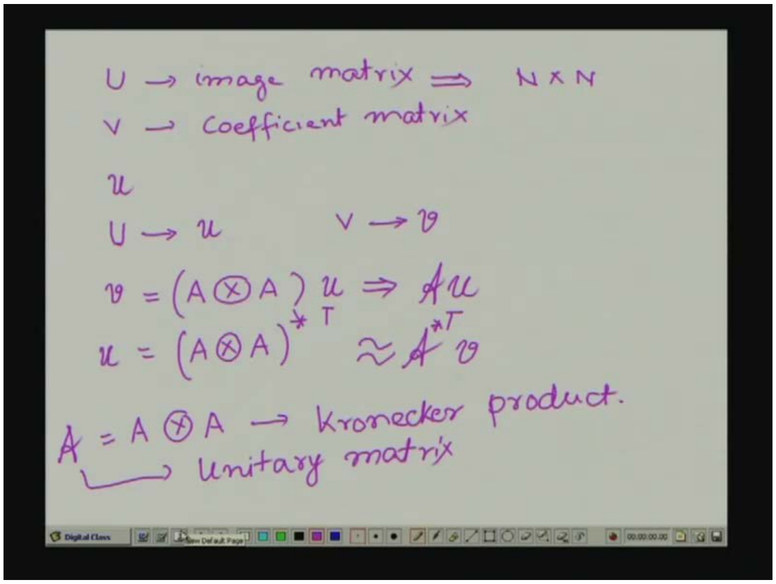
(Refer Slide Time: 44:18)



Now, to see that what will be the result of inverse transformation, you remember the transformation coefficient matrix V, we had obtained as (5, minus 1, minus 2 and 0). So, this was our coefficient matrix. By inverse transformation, what we get is or inverse transformation is A conjugate transpose VA conjugate which by replacing these values, we will get as half into (1, 1, 1, minus 1) then (5, minus 1, minus 2, 0) and again (1, 1, 1, minus 1) and if you compute this matrix multiplication, the result will be (1, 2, 3, 4) which is nothing but our original image matrix U.

So, here again, you will find that by the inverse transformation, we get back our original image U and we have also found that what are the basis images, the 4 basis images - A star $_{0,0}$, A star $_{0,1}$, A star $_{1,0}$ and A star $_{1,1}$ for this particular transformation matrix A which has to be operated on the image matrix U and we have also seen that by the inverse transformation, we can get back the original image matrix U. Now, let us look further in this separable transformation.

(Refer Slide Time: 46:04)



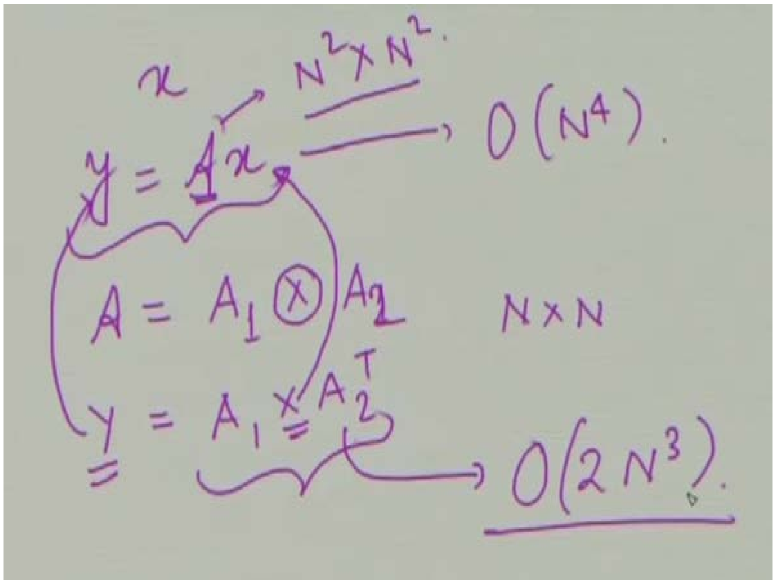
So, what we had in our case is we had U as the original image matrix and after transformation, we get V as the coefficient matrix and you would remember that both these matrices are of dimension capital N by capital N. Now, what we do is for both these matrices U and V, we represent them in the form of vectors by row ordering. That is we concatenate one row after another. So, by this row ordering, what we are doing is we are transforming this matrix of dimension capital N by capital N to a vector of dimension capital N square and by this row ordering, the vector that we get let us represent this by the variable say u.

So, by row ordering, the input image matrix is mapped to a vector say u. Similarly, by row ordering, the matrix coefficient, matrix V is also represented by v. Now, once we do this, then this transformation equations can also be written as v is equal to A Kronecker product with Au. So, this Kronecker product of A and A can be represented as this A and it is represented by A into u.

Similarly, the inverse transformation can also be written as u is equal to A Kronecker product of A conjugate transpose which is nothing but A sorry A conjugate transpose v where this particular sign A, A this represents Kronecker product and the matrix A which is equal to the Kronecker product of the 2 matrices A and A, this is also a unitary matrix.

So, once we do this, then you will find that our 2 dimensional transformation; after doing this row ordering of the input image U and the coefficient matrix V, once they are represented as 1 dimensional vectors of dimension capital N square, so this 2 dimensional image transformation is now represented in the form of or in a 1 dimensional transformation form.

(Refer Slide Time: 49:48)



So, by this what we have is say, any arbitrary 1 dimensional signal say x can now be represented as, say y can now be transformed as y equal to Ax and we say that this particular transformation is separable where A is the transformation matrix; we say that this transformation is separable if this transformation matrix A can be represented by as the Kronecker product of 2 matrices A_1 and A_2.

So, whenever this transformation matrix A is represented as Kronecker product of 2 matrices, A 1 and A_2 sorry A_2, then this particular transformation is separable because in this case, this transformation operation can be represented as y equal to A_1 x into A_2 transpose where this y is the coefficient matrix and x is the input matrix and we have mapped this y into a vector y by row ordering and this matrix x is mapped into this vector x again by row ordering.

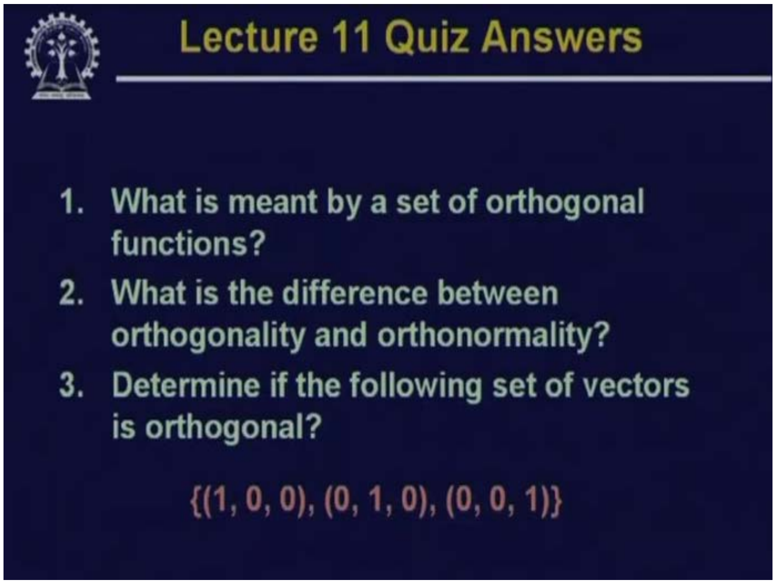
Now, if we represent this in this form, then it can be shown that if both A_1 and A_2 are of dimension N by N and then because this A is the Kronecker product of A_1 and A_2; this A will be of dimension N square and by this matrix multiplication, again we can see this will be of dimension N square by N square.

So, total N to the power of 4 numbers of elements. So, the amount of computation that you have to do in this particular case will be again of order N to the power 4 and because this transformation A is separable and this can be represented as Kronecker product of A_1 and A_2 and you will find that this particular operation can now be obtained using N cube number of operations order N cube number of operations.

So, this again says that if a transformation matrix is represented as Kronecker product of 2 smaller matrices, then we can reduce the amount of computation. So obviously, if both A_1 and A_2 can be further represented as Kronecker product of other unitary matrices, then it is possible that we can reduce the computation time further and effectively actually that is what is done in case of fast transformations.

So, today we have discussed about the separable transformation and we have seen that how this separable transformation can be used to reduce the computational complexity.

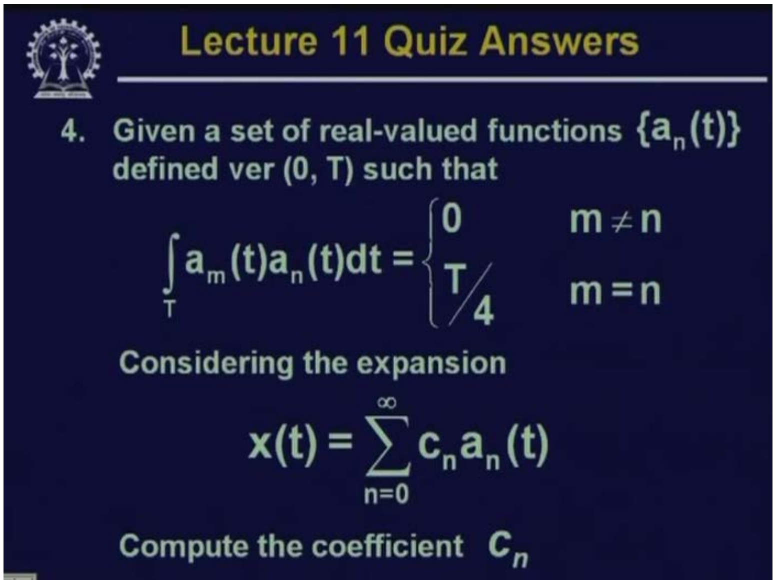
(Refer Slide Time: 53:05)



Now, the answers to the quiz questions that we given in the last class. Obviously, in this case the first 2 are quite obvious. The third one which we said that determine if the following set of vectors is orthogonal. Now, what you have to do is you have to check whether these vectors, these 3 vectors are pair vise orthogonal or not.

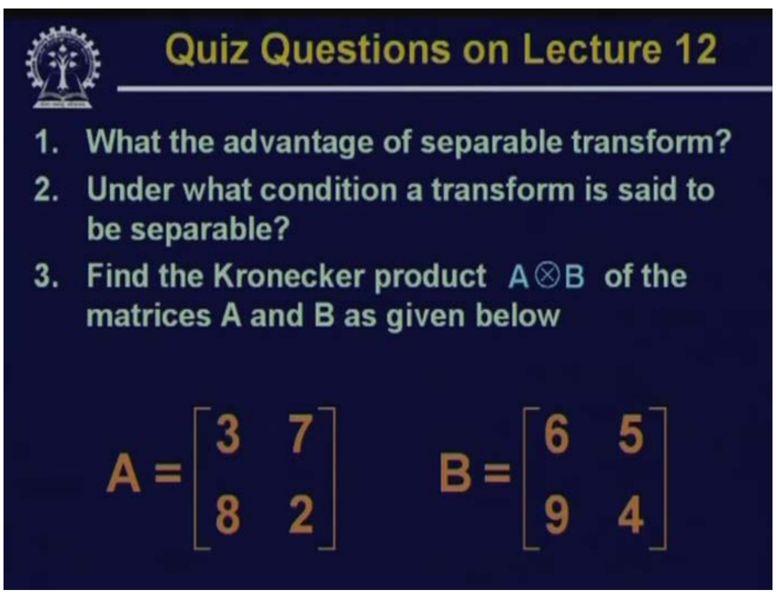
That is if you take the inner product of pair of these vectors, then only if you take the inner product of the vector with itself, you should get a non 0 value and if you take the inner product of 2 different vectors, you should get a 0 value and you will find that if you verify on this, you will get the same result that is $A_{1, 0, 0}$, inner product with 1, 0, 0 that will be 1. But 1, 0, 0, 1 inner product with 0, 1, 0 or inner product with 0, 0, 1 that will be 0 and you can verify that obviously this particular set of vectors is orthogonal and it is not only orthogonal, this particular set of vectors will be orthonormal.

(Refer Slide Time: 54:21)



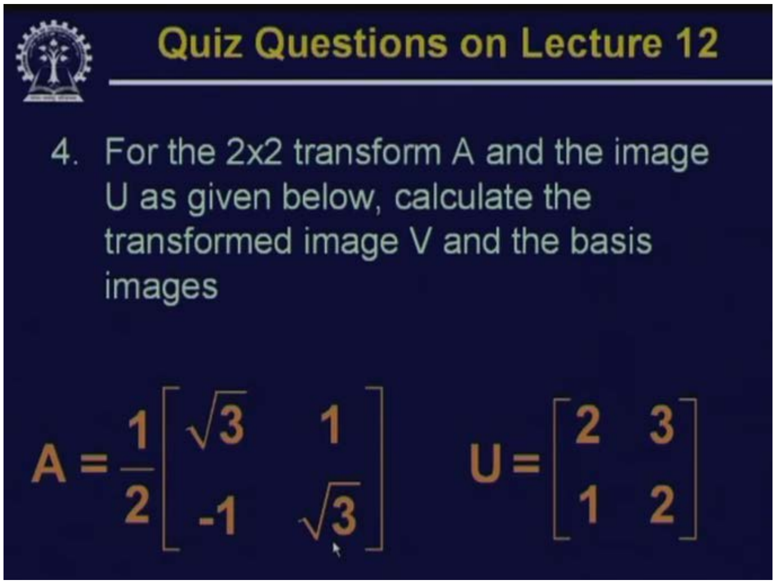
The fourth one, you have to find out the coefficient C_n. Obviously this comes from the definition that if you integrate x (t) a_n (t) dt over the interval capital T; then that gives you the value of this coefficient C_n. So, this again straight way comes from the lecture material that we have covered in our previous class that is lecture number 11.

(Refer Slide Time: 54:53)



Now, coming to today’s quiz questions; the first question is what is the advantage of separable transform? Second question: under what condition, a transform is said to be separable? The third question: here we have given 2 matrices A and B, you have to find out the Kronecker product of the matrices A and B.

(Refer Slide Time: 55:27)



The fourth one: here we have given 2 matrices, the first one is the transformation matrix A and the second one is the input image matrix U. You have to calculate the transform, this image matrix U when transformed with the transformation matrix A and you also have to find out the corresponding basis images.

Thank you.