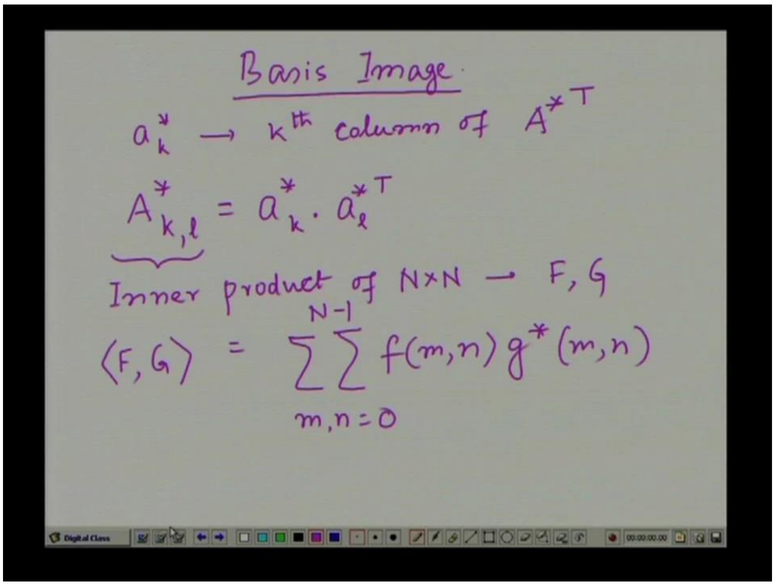
Digital Image Processing.
Professor P. K. Biswas.
Department of Electronics and Electrical Communication Engineering.
Indian Institute of Technology, Kharagpur.
Lecture-23.
Basis Images.

(Refer Slide Time: 0:37)

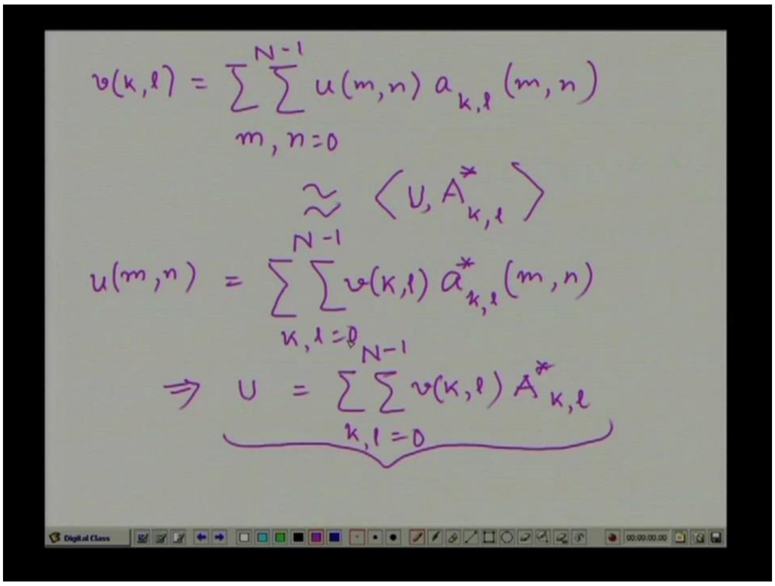


Hello, welcome to the video lecture series on digital image processing. Now let us see that what is meant by the basis images. So what is meant by basis image. Now here we assume that suppose ak* this denote the kth column of the matrix A conjugate transpose. So ak* we represent this the kth column of A conjugate transpose, where A is the transformation matrix.

And now if I define the matrices Ak,l* as ak* into al* transpose. So we find that ak* is the kth column of the matrix a* transpose, al* is also the lth column of the matrix A conjugate transpose. So if I take the product of ak* and al* transpose, then I get the matrix a matrix Ak,l*, ok. And let us also define the inner product of say two N by N matrices, so I define inner product of two N by N matrices, say F and G.

So it the inner product of these two matrices <F and G> are defined as f(m,n)g*(m,n), where both m and n vary from 0 to capital N-1. So we define the inner product of two matrices <F and G> in the form of f(m,n)g*(m,n) where both m and n vary from 0 to capital N-1.

(Refer Slide Time: 3:16)



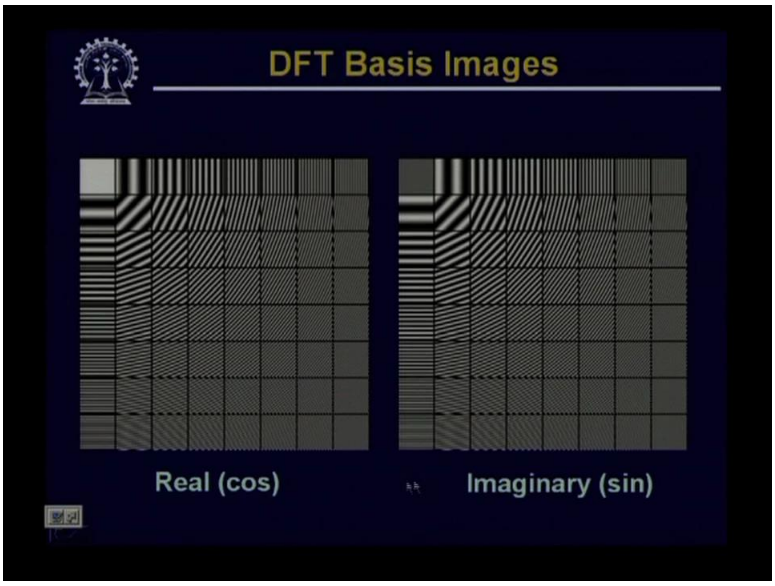
So now by using these two definitions now if I rewrite our transformation equations, so now we can write the transformation equations as v(k,l) is equal to you find that the old expression that we have written u(m,n) ak,l(m,n), where both m and n vary from capital 0 to capital N-1.

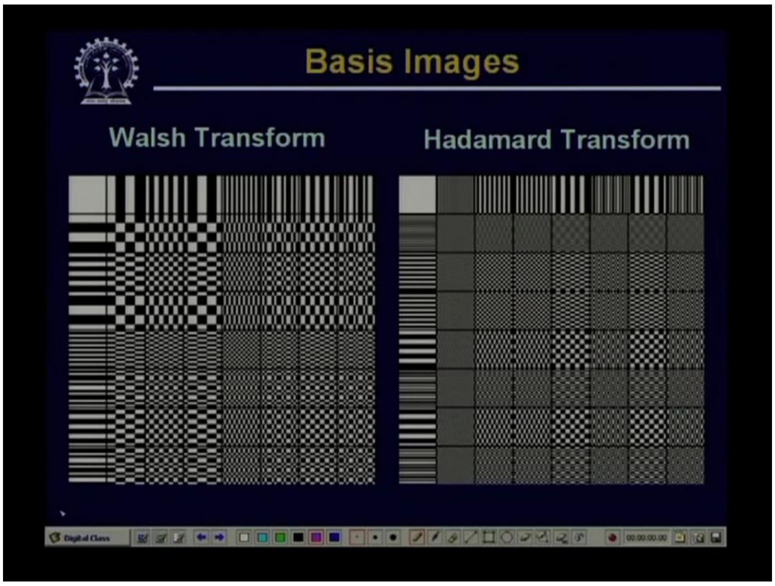
So this is nothing but as per our definition so if you just look at this definition. This is nothing but an expression of an inner product. So this was the expression of the inner product. So this transformation equation is nothing but an expression of an inner product and this inner product is the inner product of the image matrix U with the transformation matrix A*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*k,l(m,n), where k,l vary from 0 to capital N-1.

So again we find that in the matrix form this will be written as U equal to summation v(k,l) into A*k,l, where both k and l vary from 0 to capital N-1. So if you look at this particular expression you find that our original image matrix now is represented by a linear combination of N square matrices A*k,l. Because both k and l vary from 0 to capital N-1, so I have N square such matrices Uhh ak,l.

And by looking at this expression you find that our original image matrix U is now represented by a linear combination of N square matrices A*k,l, where each of this N square matrices are of dimension Uhh capital N by Capital N. And these matrices A*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.

(Refer Slide Time: 6:49)





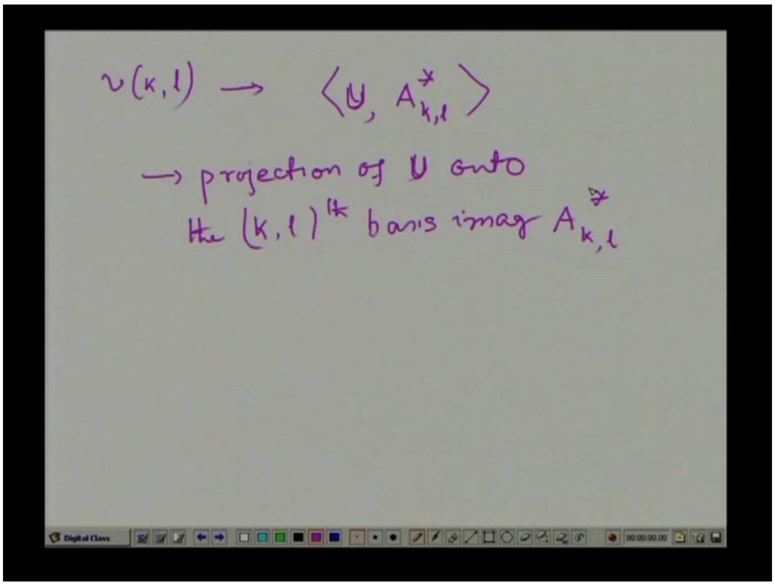
Now to look at how these basis images look like, to see how these basis images look like. Let us see some of the images, so here we find that we have shown two images we will see later that these are the basis images of dimension 8 by 8. So here we have shown basis images of dimension 8 by 8. And there are total 8 into 8 that is 64 basis images. We will see later that in case of discrete fourier transformation we get two components one is the real component, other one is the imaginary component.

So accordingly we have to have two basis images one corresponds to the real component, other one corresponds to the imaginary component. Similarly this is another basis image which corresponds to the discrete cosine transformation, so again here I have shown Uhh the basis images of size N by Uhh 8 by 8, ofcourse the image size, image is quite expanded and again we have 8 into 8, that is 64 number of images.

So here we find that a row of this Uhh 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. 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 these basis images will be weighted by the corresponding coefficient the transformation coefficient (vk,l) that we compute after the transformation. And as we have said that this (vk,l) is nothing but the inner product of (k,l)th basis image. So when we compute this (vk,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 A*k,l.

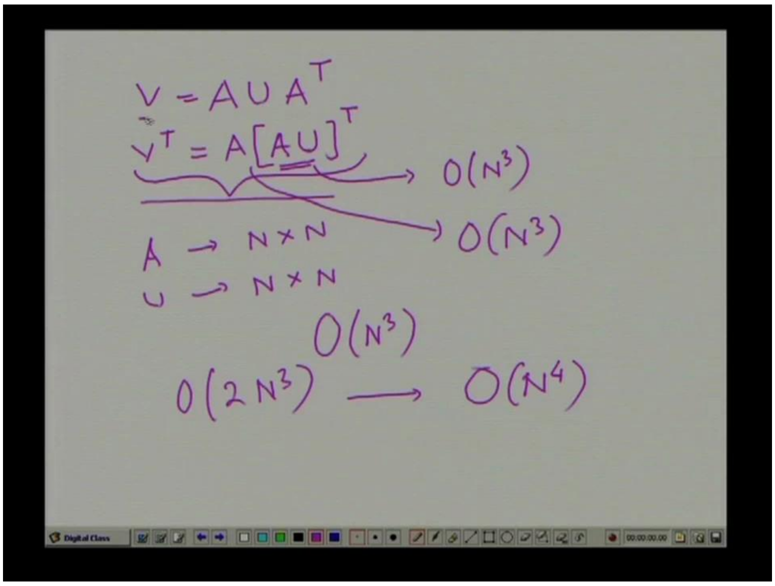
(Refer Slide Time: 9:50)

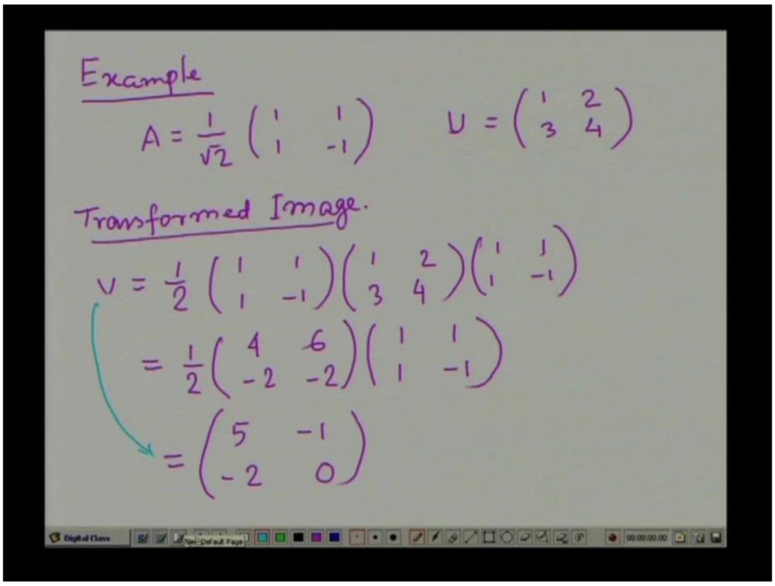


So each of these coefficients v(k,l) is actually represented as the inner product of the input image U with the (k,l)th basis image Ak,l*. And because this is the inner product of the input image U and the (k,l)th basis image Ak,l*, this is also called the projection of the input image on the (k,l)th basis image. So this is also called the projection of the input image u onto the (k,l)th basis image Ak,l*, ok.

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 Uhh of the image transformation. So let us take a an example.

(Refer Slide Time: 11:22)



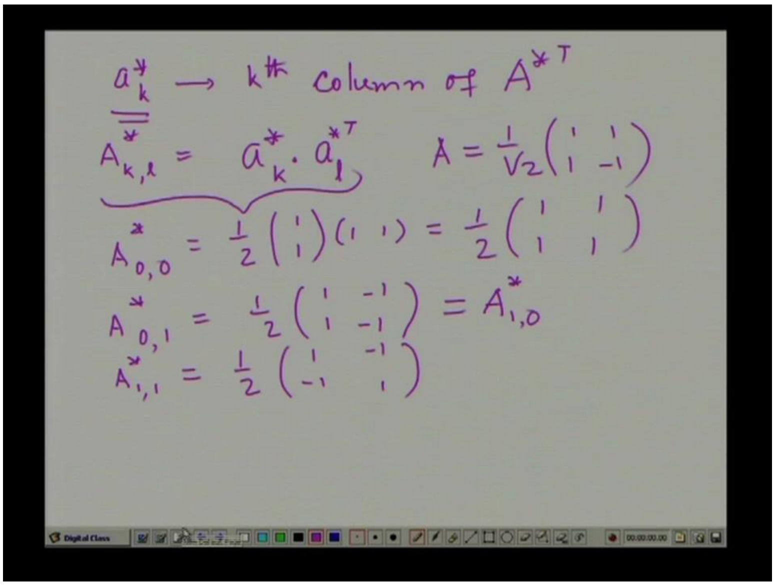


So let us consider an example of this transformation. Say we have been given a transformation matrix which is given by A equal to 1 upon root 2,(1,1,1,-1) and we have the input image matrix U equal to (1,2,3,4).

And in this example 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,-1) into our input image (1,2,3,4).

So if you just see our expressions, you find that our expressions was something like this. When we computed V, we had computed V equal to AUA transpose, ok. So by using that we have A, U then A transpose and by nature of this transformation matrix A, you find that A transpose is nothing but same as A. So will have (1,1,1,-1). And if you do this matrix computation it will simply come out to be 1 upon 2 into (4,6,-2,-2) into (1,1,1,-1) and on completion this matrix multiplication the final coefficient matrix V will come out to be (5,-1,-2,0). So I get the coefficient matrix V as (5,-1,-2,0), ok.

(Refer Slide Time: 14:34)

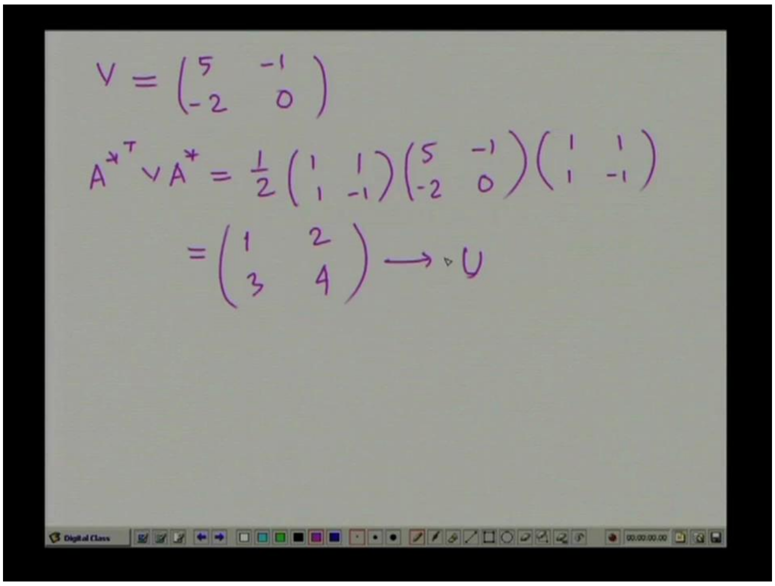


Now let us see that what is the for this particular transformation what will be the corresponding basis images. Now when we define the basis images, you remember that we have said that we have assumed ak* to be the kth column, this was the kth column of matrix A* transpose, ok. Now using the same concept and form this our basis functions was taken as Ak,l* which was given by ak* multiplied with Uhh sorry ak,l* the k,lth basis image was computed as ak*multiplied with al* 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 uhh the transformation matrix A is given as 1 upon root 2 (1,1,1,-1), I can compute the basis images as A0,0*, the 0th 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 A0,1 that is 0,1th basis image will be given as half into (1,1,-1,-1) which will be same as A1,0, that is 1,0th basis image and similarly we can also compute A1,1 that is 1,1th basis image will come out to be half into (1,-1,-1,-1). So this is simply by the matrix Uhh multiplication operations we can compute these basis images from the rows of Uhh from the columns of A conjugate transpose.

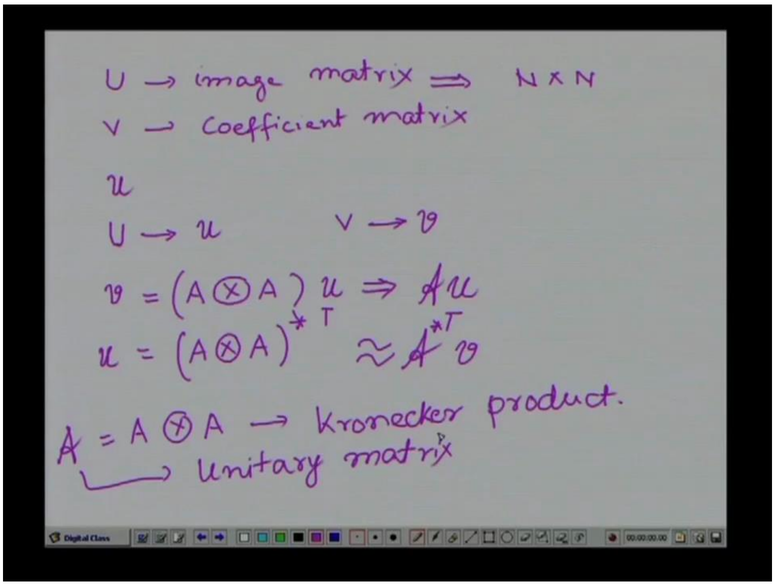
(Refer Slide Time: 17:16)



Now to see that what will be the result of inverse transformation, you remember the transformation coefficient matrix V we had obtained as (5,-1,-2 and 0). So this was our coefficient matrix. By inverse transformation what we get is our inverse transformation is A conjugate transpose VA conjugate which by replacing these values will get as half into (1,1,1,-1) then (5,-1,-2,0) and again (1,1,1,-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 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 four basis images A*0,0, A*0,1, A*1,0 and A*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.

(Refer Slide Time: 18:56)



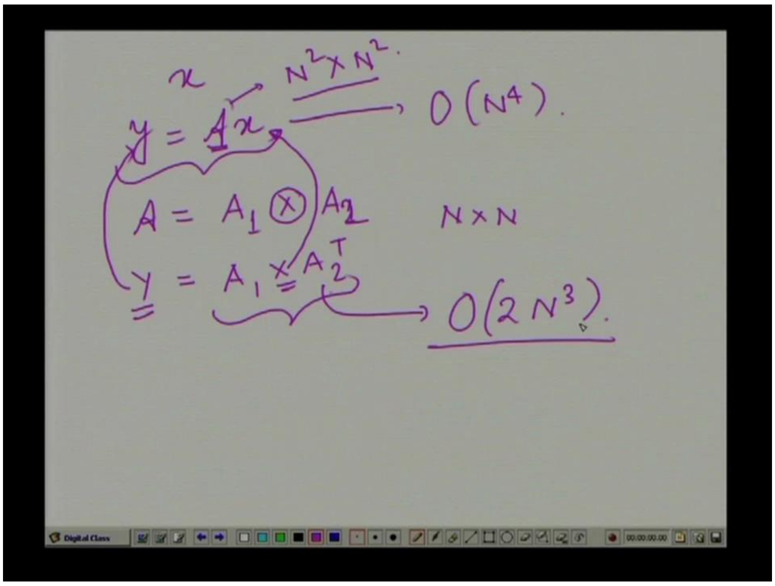
Now let us look further in this Uhh separable transformations. So what we had in our case is we had U as the original image matrix and after transformation we get V as the coefficient matrix. And you would remember that both these matrices are of dimension capital N by capital N. Now what we do is for 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 Uhh 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 A u, ok. 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 two matrices A and A this is also an unitary matrix. So once we do this, then you find that our two dimensional transformation after doing this row ordering of the input image U and the coefficient matrix V, Uhh once they are represented as one dimensional vectors of dimension capital N square, so this two dimensional image transformation is now represented in the form of Uhh in a one dimensional transformation form.

(Refer Slide Time: 22:49)



So by this what we have is say any arbitrary one 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 two matrices A1 and A2.

So whenever this transformation matrix A is represented as kronecker product two matrices of A1 and A2, sorry A2, then this particular transformation is separable. Because in this case this transformation operation can be represented as Y equal to A1 X into A2 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 A1 and A2 are of dimension N by N and then because this A is the kronecker product of A1 and A2, this A will be of dimension N square, ok. And by this matrix multiplication again we can see Uhh this will be of dimension N square by N square so total N to the power 4 number of elements. So the amount of computation that we 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 A1 and A2 and you 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 two smaller matrices then we can reduce the amount of computation, thank you.