Information Theory and Coding

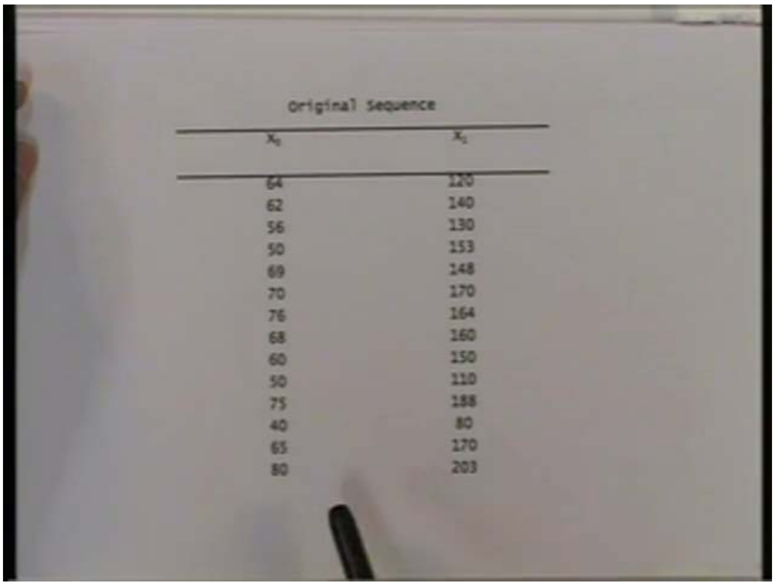
Prof. S. N. Merchant
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture - 40
Transform Coding - Part – I

Vector quantization is a block quantization approach. In this approach we exploit the correlation between the samples of the source output vector of a certain length k, in order to place the output points of the quantizer in appropriate position in k dimensional space. In today's class, we will study yet another block quantization approach. In this approach, we group the source samples in form of a vector of a certain length. And then use a linear reversible transform to transform this vector into components that are then quantized according to the individual characteristics. This approach is known as transform coding.

Before we examine the mathematical delicacy of transform coding. Let us try to understand the working principle of transform coding, and appreciate its advantage with the help of a couple of examples. Let us start with an example where a source generates a sequence of pairs of numbers. Now from this sequence we form source output vectors of length 2 by associating the first component of this vector with the first number of this pair and denoted by x 0. And the second component of this vector is associated with the second number of this pair and denoted by x 1, so a typical sequence from such a source is shown here.

(Refer Slide Time: 03:18)

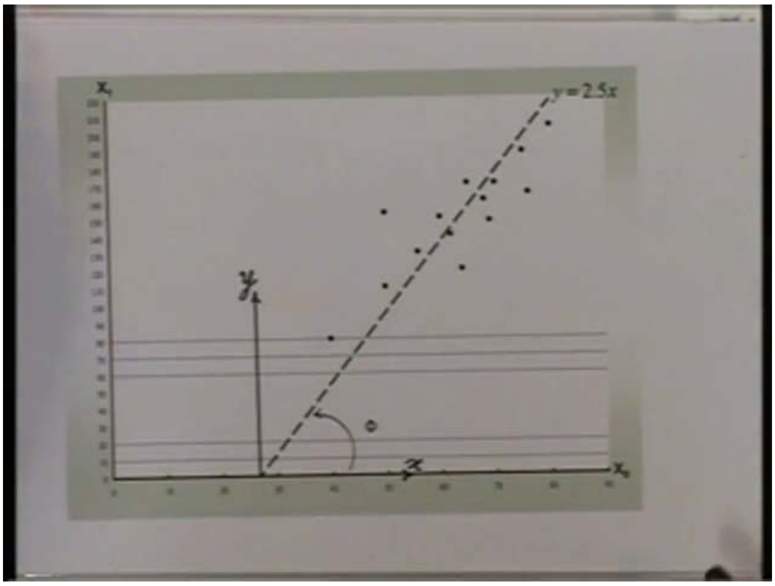


Original Sequence

x_0	x_1
64	120
62	140
56	130
50	153
69	148
70	170
76	164
68	160
60	150
50	110
75	188
40	80
65	170
80	203

Now, each of this source output vector can be represented as a dot in two dimensional space, where the horizontal axis represents x 0 and the vertical axis represents x 1. Now, if we do this we get the following plot.

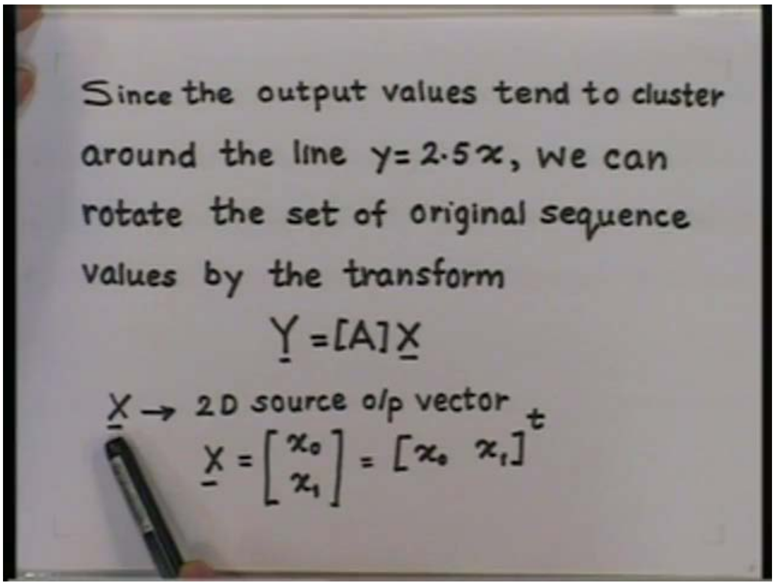
(Refer Slide Time: 03:47)



Now, let us make some observations about this plot first in the coordinate space (x, y) as shown here, where x axis align with x 0 axis then in this x y coordinate space. We find that this dots which represent the source output vectors cluster around the line y is equal to 2.5 x. Another observation from this plot is that horizontal coordinate is correlated with the vertical coordinate. Another observation is that except for the different means, the distribution of the horizontal coordinate is very similar to the distribution of the vertical coordinate, and the variance of the horizontal coordinate is very much similar to the variance of the vertical coordinate. Now, as we have studied earlier that entropy at least for Gaussian source is directly proportional to the variance. So, what this plot indicates is that the entropy which is a measure of information for the horizontal coordinates is similar to the entropy of the vertical coordinate, because horizontal coordinate and vertical coordinate are correlated.

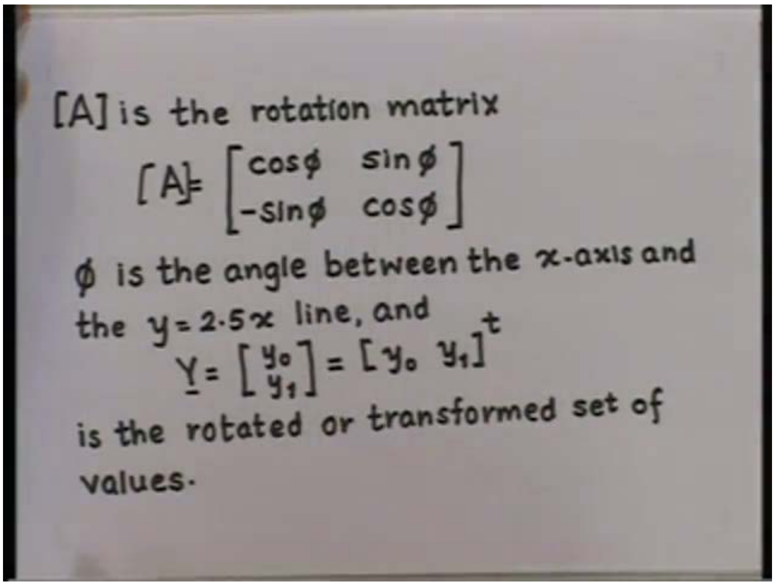
What this implies is that if we quantize the horizontal coordinate, and vertical coordinate separately then the outputs of these quantizers will have lot of common information. This will to be an efficient way of transmitting the pairs of this samples, or storing this pair of the samples, so what should be done? So, one approach is that we take this coordinate space denoted by (x 0, x 1) and rotate it about (0, 0) of (x, y) coordinate space by angle phi.

(Refer Slide Time: 07:07)



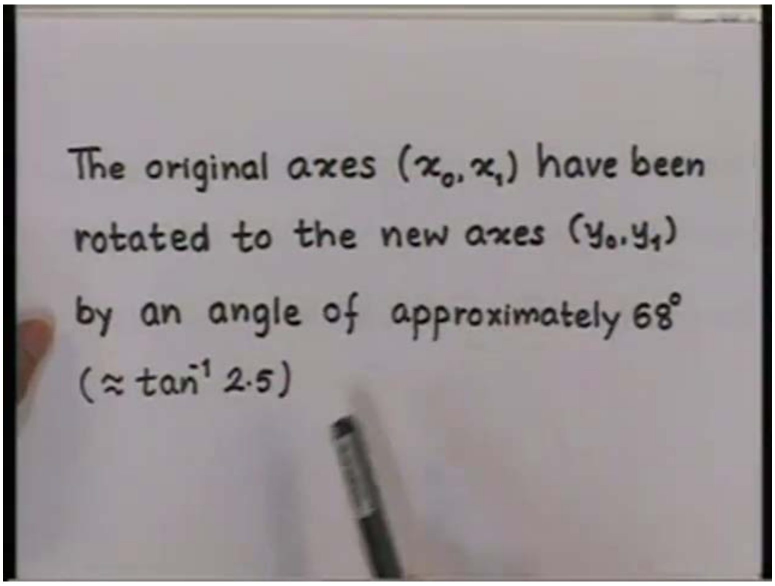
Now, if we do this then this is equivalent to the rotation of the set of original sequence value by the transform Y is equal to A X, where X denotes 2 D source output vector.

(Refer Slide Time: 07:23)



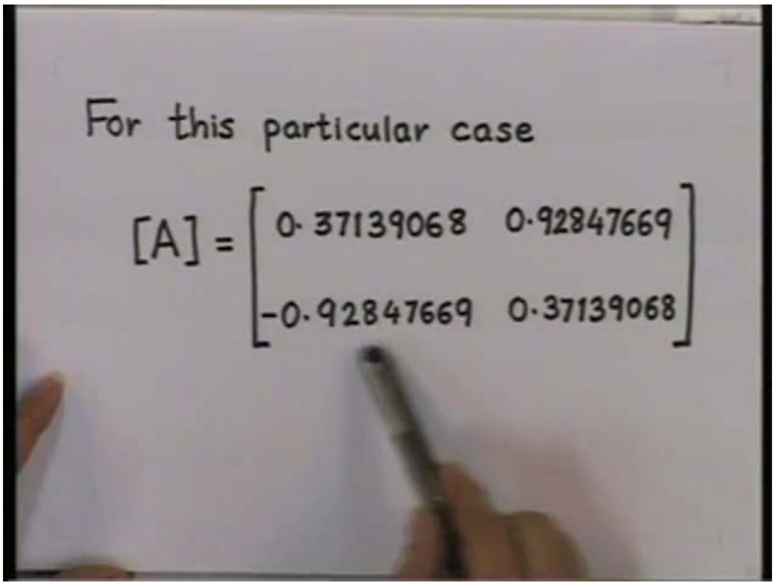
And A is the rotation matrix, where phi is the angle between the x axis and the y equal to 2.5 x line and y is the rotated or transformed set of values.

(Refer Slide Time: 07:58)



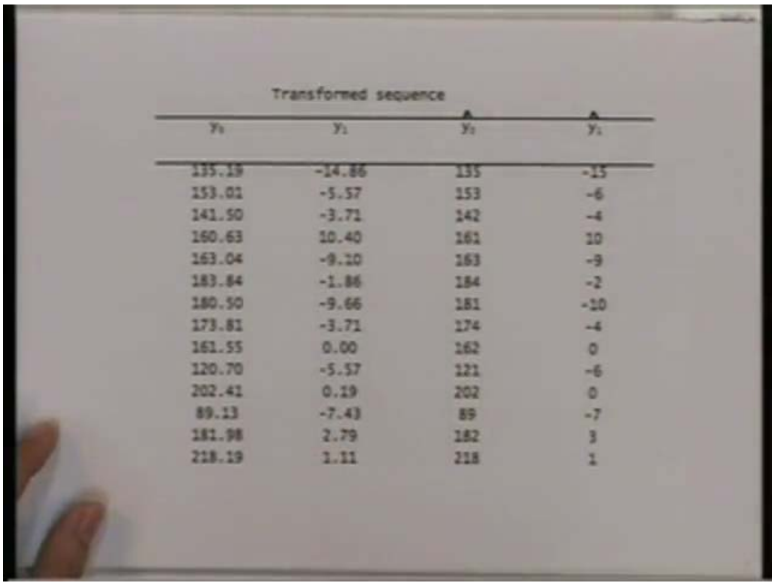
Now, in this case the original axis (x 0, x 1) has been rotated to the new axis (y 0, y 1) by an angle of approximately 68 degrees which is tan inverse 2.5.

(Refer Slide Time: 08:18)



In this case we get the rotation matrix a given here, now once we have the rotation matrix, we can transform it and obtain the output as shown here.

(Refer Slide Time: 08:35)



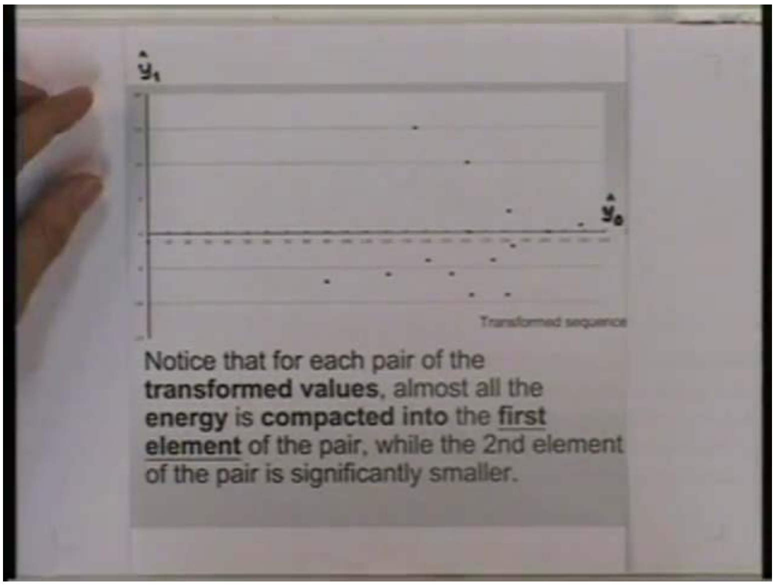
Transformed sequence

y_0	y_1	$\hat{y}_0$	$\hat{y}_1$
135.19	-14.86	135	-15
153.01	-5.57	153	-6
141.50	-3.71	142	-4
160.63	10.40	161	10
163.04	-9.10	163	-9
183.84	-1.86	184	-2
180.50	-9.66	181	-10
173.81	-3.71	174	-4
161.55	0.00	162	0
120.70	-5.57	121	-6
202.41	0.19	202	0
89.13	-7.43	89	-7
181.98	2.79	182	3
218.19	1.11	218	1

So, we get the output y 0 and y 1, now the next step in transform coding is to quantize this outputs y 0 and y 1. Let us assume that we use the same quantizer for both the components y 0 and y 1, we use a uniform quantizer and quantize it to the nearest integer. If we do that then what we get is the following result, so y 0 cap is the quantization of the values y 0 to the nearest integer and y 1 cap is the quantization output of y 1 to the nearest integer. Now, from this y 0 cap and y 1 cap, we can obtain back the original sequence to some approximation by using an inverse transform in the form of a inverse.

Now, if we did that then we are not gaining any real advantage from the transform coding except for the fact that earlier we would have quantized x 0 and x 1 sample, but now we have quantized y 0 and y 1 samples. So, if you want to gain the real advantage from the transform coding then in that case we should exploit the characteristics of y 0 and that of y 1. Now, in order to study this characteristic we could plot these points in (y 0, y 1) space, but now since y 0 cap is very close to y 0 and y 1 cap is very close to y 1, for the sake of convenience we plot the points in y 0 cap and y 1 cap space.

(Refer Slide Time: 11:17)

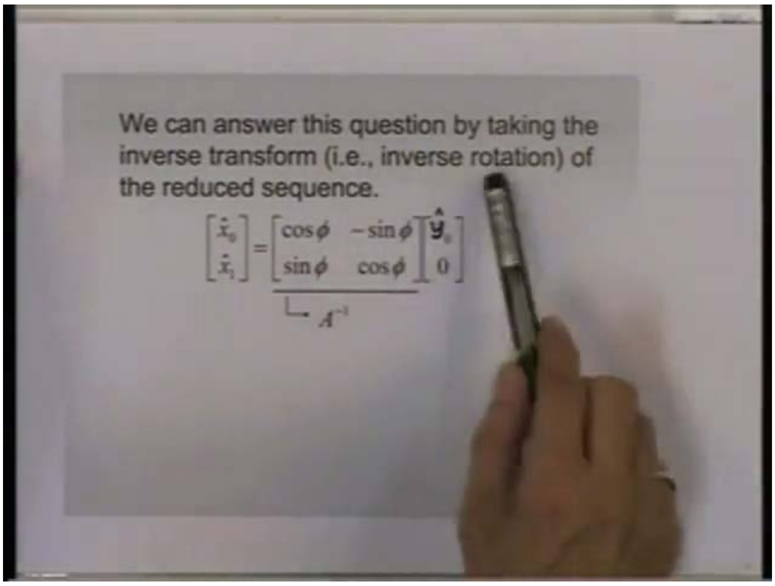


So, on doing so we get this plots in y 0 cap and y 1 cap space now the characteristic depicted by this plot very well resembles the characteristics of the points in (y 0, y 1) space. And from this plot we can observe the following if we define the energy as the sum of the squares of the horizontal coordinate plus the square of the vertical coordinate. Then this plot shows that almost all the energy is compacted into the first element of the pair indicated by y 0, while the second element of the pair indicated by y 1 is significantly smaller. Another observation is that that now y 0 and y 1 are decode related, another observation is that that distribution for y 0 is very much different from that y 1 the variance of y 0 is much larger than the variance of y 1.

Therefore, as discussed earlier the information contained in y 0 is much higher than the information contained in y 1 and now since y 0 and y 1 are decode related there is very less common information between the two components. Now, as said earlier that if we really want to get the advantage of transform coding then we should use this characteristic to quantize y 0 and y 1.

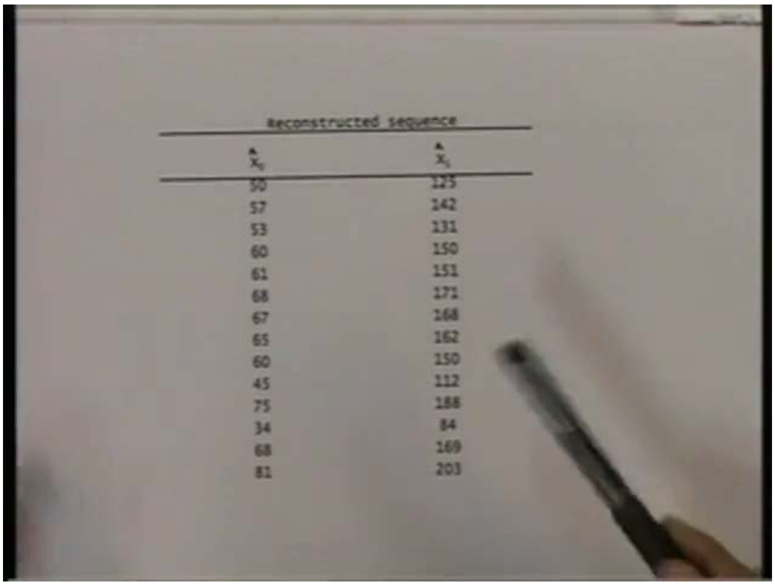
So, based on this result depicted by this plot let us do the following, let us quantize y 0 as y 0 cap by using a uniform quantizer which quantizes to the next nearest integer and get y 1 cap as the quantizer output for y 1 which is all 0. Now, what this means that we have set all the second elements of the transformation of y 1 to 0, now by doing this number of elements that need to be coded reduces by half. Now, the question is what is the effect the throwing away half the elements of the sequence and now we can answer this question by taking the inverse transform.

(Refer Slide Time: 14:44)



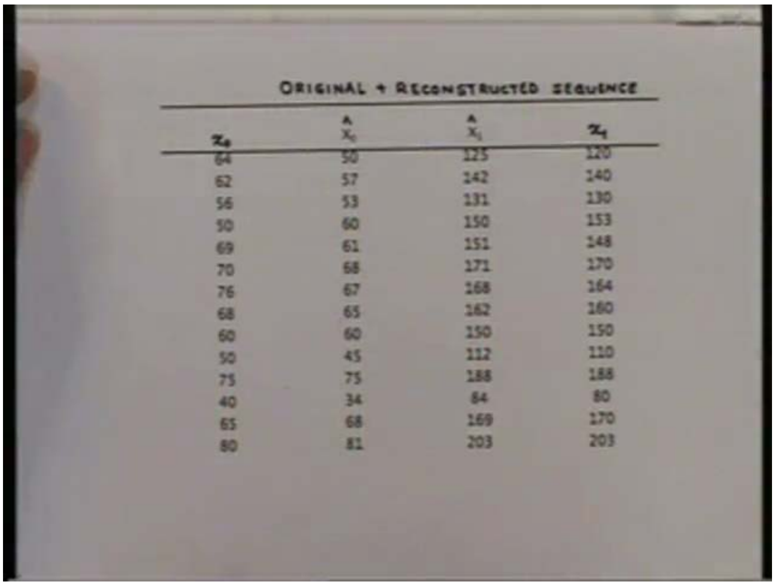
That is inverse rotation of the reduced sequence, so x 0 cap and x 1 cap denote the reconstructed sequence or the approximated original sequence which is obtained by a inverse operating on y 0 cap and 0, because y 1 cap has been reduced to 0. If we do this the approximated original sequence, or the reconstructed sequence is shown out here.

(Refer Slide Time: 15:20)



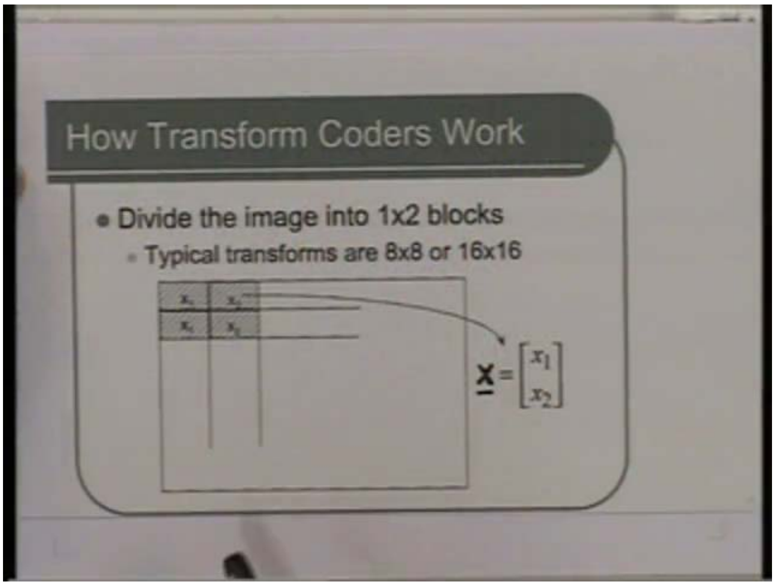
And if we compare this reconstructed sequence with the original sequence.

(Refer Slide Time: 15:35)



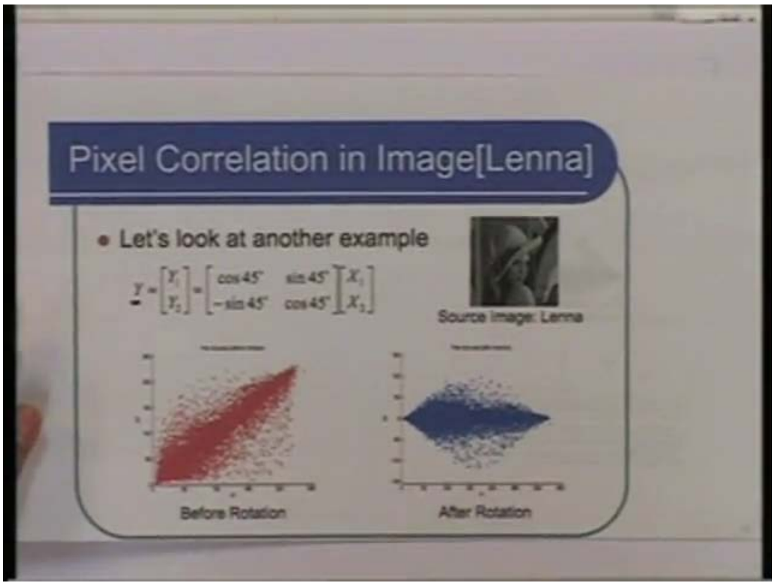
This is the table which depicts that, so this is the original sequence and this is the approximated, or reconstructed original sequence after we throw away half the number of components of the transformed coefficient. Similarly, this is the reconstructed second component of the sequence and this is the original second component of the sequence. Now, comparing the reconstructed sequence with the original sequence, we see that even though we transform only half the number of elements presented in the original sequence, the reconstructed sequence is very close to the original sequence. Now, let us take another example from, now let us take another example where the source is an image. So, let us see how the transformed coding works on images, so the first step is to obtain the source output vector and this is done as follows.

(Refer Slide Time: 16:53)



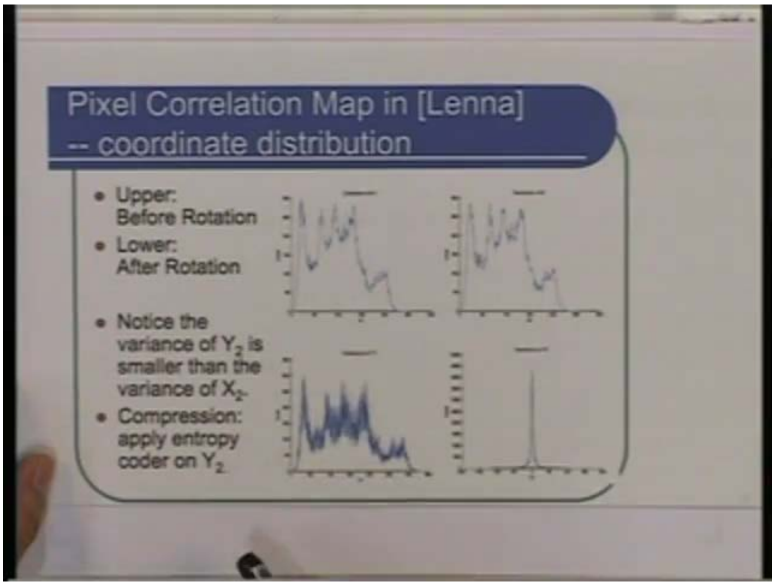
For our discussion we form source output vectors of length 2 by dividing the image into 1 by 2 block though in practice typical transforms will use the block size of 8 by 8, or 16 by 16. So, this vector x will be either of size 64 or 256 and it is obtained by scanning this blocks from left to right and from top to bottom. So, for our discussion we will assume that we have 1 by 2 blocks, so the vectors are of length 2, now using this principle.

(Refer Slide Time: 17:47)



If we take the source image as Lenna and plot the source output vectors as shown here with x 1 representing the horizontal axis and x 2 representing the vertical axis. Then this is the distribution of the source output vectors that is the distribution of adjacent pixels from this plot, it is very clear that most of this points cluster around the point x 2 equal to x 1 naught. Let us see what happens if we rotate this axis by 40 degrees by using this transform, the difference between this transform the earliest transform is that now we have 45 degrees instead of 68 degrees. And when we do this and we plot the transform components y 1, y 2 space this is the plot which we get. Now, let us study the characteristics of this distribution before rotation and after rotation.

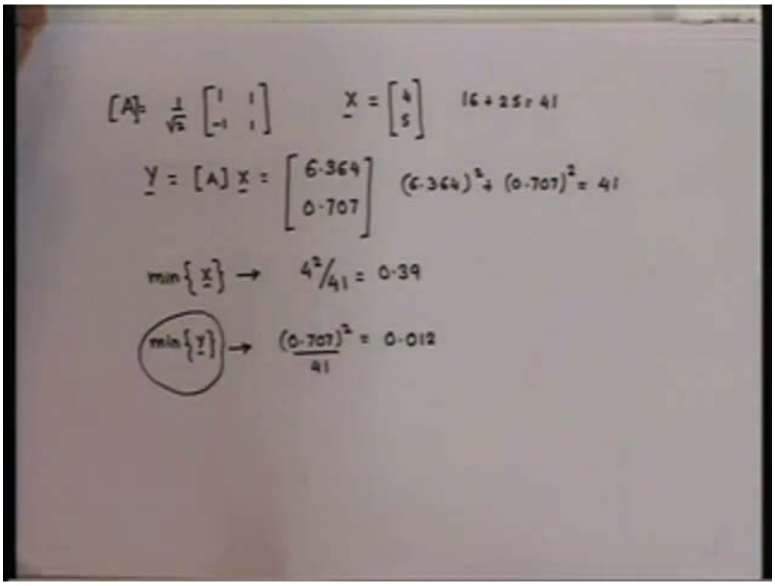
(Refer Slide Time: 19:14)



If we look at the distribution of x 1 before rotation, it is given here and if you look at the distribution of x 2 before rotation it is given here and so from this distribution we find that the information contained in this and in this distribution is more or less similar. And because x 1 and x 2 are correlated as depicted by this plot, it implies that there is lot of common information the adjacent pixels, so it will not be wise for us to quantize them separately. Now, on rotation for this plot if you look at the distribution of y 1 this is what we get and if you look at the distribution of y 2, this is what we get. So, again we note that the variance of y 2 is much smaller than the variance of x 2.

So, if we adopt a similar strategy what we did in the earlier example and throw off all the y 2 components and reconstruct the image by using just y 1 components, we would get the similar results. So, the reconstructed output image will be very close to the original image, this is because the variance of the second component is very small when we discarded it. So, in both this example what this rotation matrix A does is basically to compact the energy into the first component.

(Refer Slide Time: 21:30)

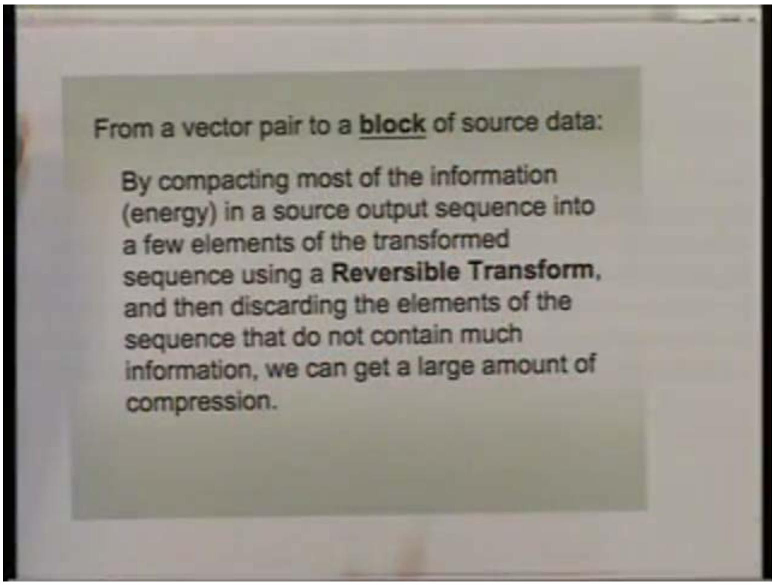


So, for example, if we choose the transform A equal to 1 by root 2, 1, 1, minus 1, 1 and have the input vector x is equal to 4, 5 to obtain the transformed vector. As now what this shows is that the energy in the x which is equal to 16 plus 25 is equal to 41 and energy in y which is 6.364 squared plus 0.07 squared is equal to again 41. So, energies are same, we will very shortly prove that this is because of the kind of transformation which we have chosen here, now what is important is that this transformation makes the second component very small.

So, in the original vector if we discarded minimum component of the vector x, we would have an error of 4 squared by 41 is equal to 0.39, but now in the transformed space if we discard the minimum component of the vector y, we have an error of 0.07 squared divided by 41 which is equal to 0.012. So, the conclusion is that we are more confident to discard the minimum of the transformed vector that the minimum component of the original vector.

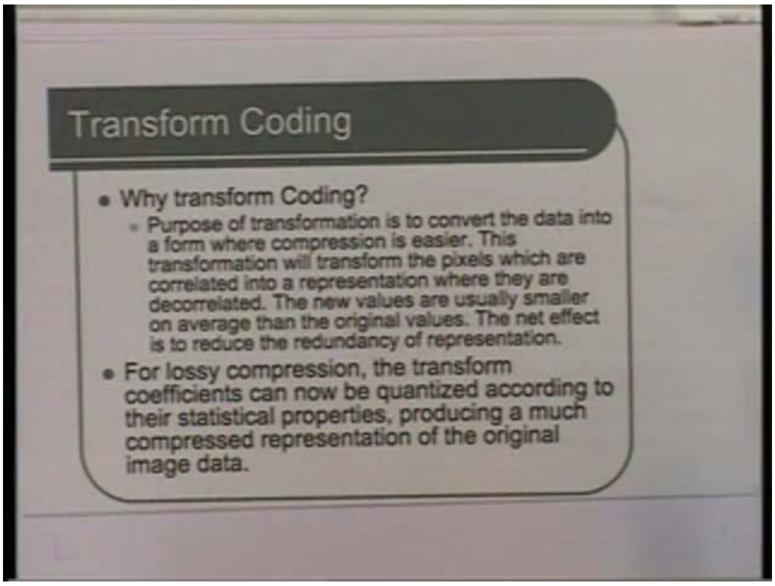
Now, in both the examples we could reduce the number of samples we needed to code because most of the information or energy containing a pair of values was put into one element of each pair because the other element of the pair contained very little information. We could discard it without significant impact on the fidelity of the reconstructed sequence. Now, in both this example we formed a vector of length 2. Now, if we exchange this idea of forming vectors of larger length by incorporating larger block of source samples then we would get the following result by compacting.

(Refer Slide Time: 25:09)



By compacting most of the information or analogy in a source output sequence into a few elements of the transformed sequence using a reversible transform. And the discarding the elements of the sequence that do not contain much information, we can get a large amount of compression, so this is the idea behind transformed coding.

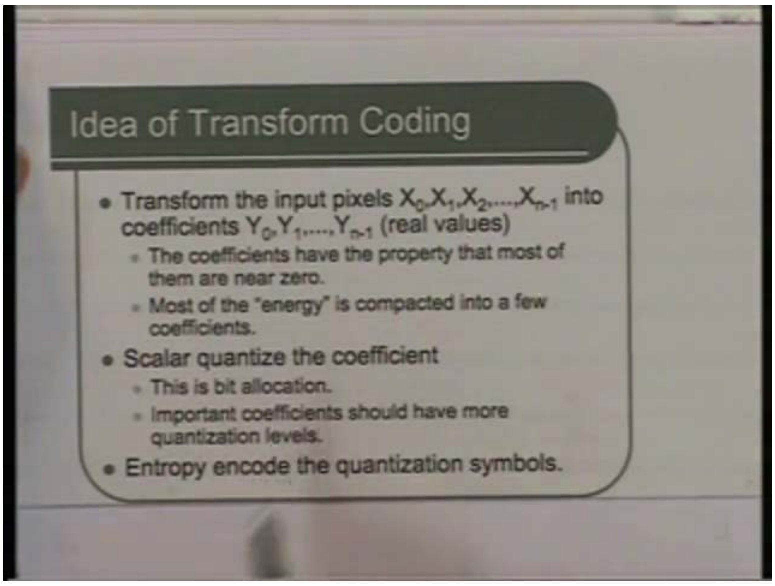
(Refer Slide Time: 25:56)



So, if we ask why use transform coding, then we can answer as follows the purpose of transformation is to convert the data into a form where compression is easier. This transformation will transform the pixels which are correlated into a relation where, they are decorrelated. So, this is very important aspect of transformed coding, the correlation existing among the samples in the source output. Vector on transformation they get decorrelated and because they get decorrelated, we can independently quantize this. So, the new values are generally, smaller than average of the new values and the net effect is to reduce the redundancy of representation because now, we have minimum amount of common information between the transformed cooperation's.

And usually, transformed coding is used for lossy compression where the transformed coefficients can now be quantized according to the statistical properties, which differ from coefficient to coefficient and in the effect producing a much compressed representation of the original image data. So, if we were to put in a nut shell the idea of transformed coding is as follows.

(Refer Slide Time: 27:41)

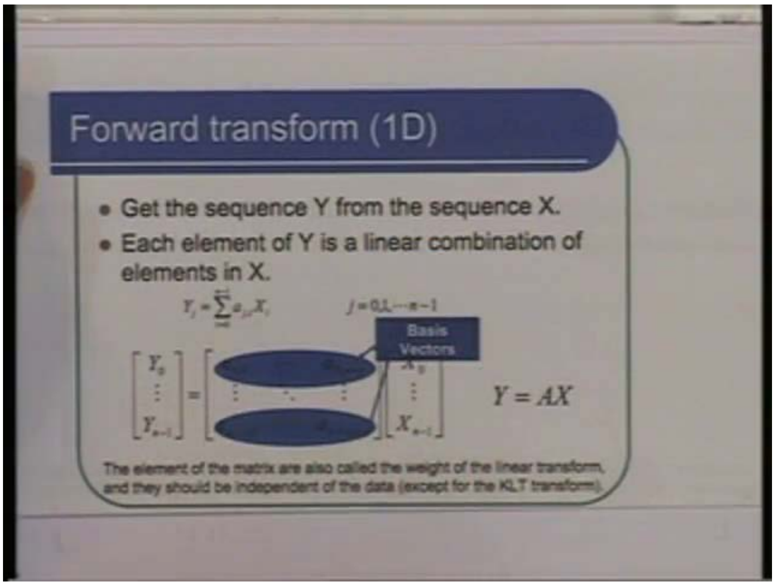


Transform the input pixels x 0, x 1, x 2 up to x n minus 1, this implies that we have made a block of size n into coefficients y 0, y 1 up to y n minus 1 without loss of generality, we will assume to be real values. Now, if we have chosen the transform to be proper then the net effect would be that the coefficients have the property that most of them are near

0. In the earlier example we saw that the second component was very near to 0, so what it means that most of the energy is compacted into a few coefficients.

Next step in transformed coding is to scalar quantize the coefficients that is do the bit allocation, that is important coefficients should have more quantization levels that is how we will carry out the transformation. And finally, the entropy encode quantization symbols and transmit it on storage at the receiver, we apply the inverse transform. So, mathematically this process of transformation can be depicted as shown here.

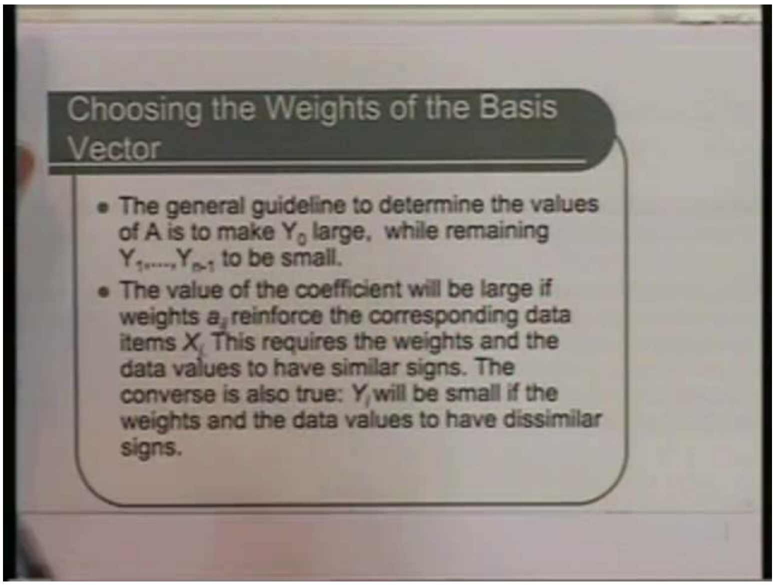
(Refer Slide Time: 29:20)



We get the sequence y from the sequence x, each element of y is a linear combination of elements in x. So, we see that the transformed values are actually the coefficients of an expansion of the input sequence in terms of the rows of the transformed matrix, so that is why the rows of this transformed matrix are often referred to as the basis vectors of the transform. The elements of the transformed sequence are often called the transform coefficients.

Now, different transforms have different basis vectors, sometimes the elements of the matrix are also called the weight of the linear transform. In practical application we want to be independent of the data except in the case of Karhunen-Loeve transform, which we will discuss very shortly. Now, the next question is how do you choose the weights of this basis vectors?

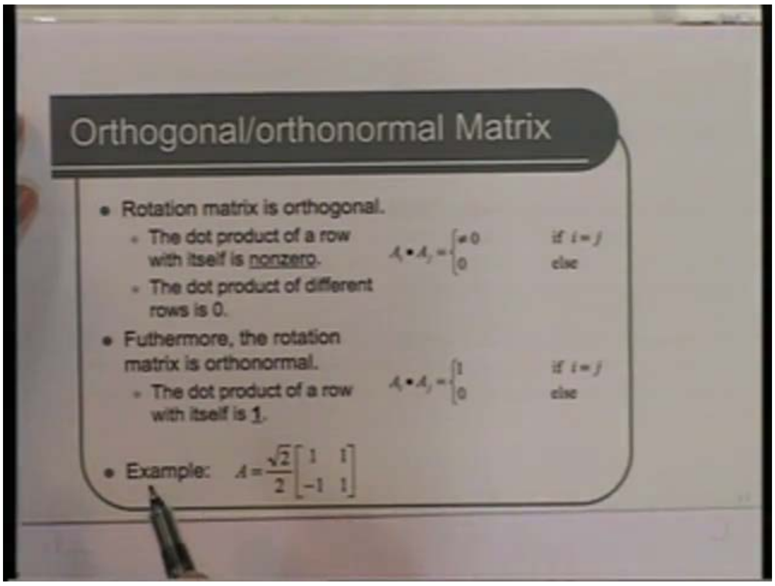
(Refer Slide Time: 30:57)



So, the general guideline to determine the values of A is to make y 0 that is then first component of the transform vector to be as large as possible while remaining y 1 up to y n minus 1 to be as small as possible. So, what this implies that the value of the coefficients will be large if weights A j, I reinforce the corresponding data items X j. So, if these weights reinforce the data items then this coefficient will be high, now this requires the weights and the data values to have similar signs. The converse is also true, Y j will be small if the weights and the data values have dissimilar signs.

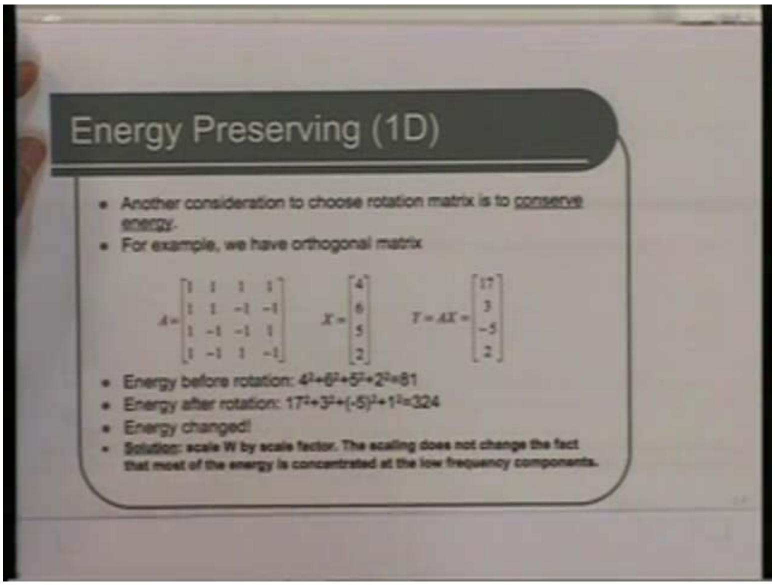
Thus, the basis vector should extract distinct features of the data vectors and must be independent and we also want it to be orthogonal, ortho-normal we will see the reason of this very shortly. Now, what this basis vectors do? Is that they pick up the low and high frequency components of this data and normally, the coefficient decrease in the order y 0 y 1 up toy n minus 1 and because of this y is now more amenable to compression than x. So, let us look at the choice of this transform in both the examples which we considered.

(Refer Slide Time: 33:28)



We find that the rotation matrix is orthogonal, what this means is that that the dot product of a row with itself is non zero, that is A i dot A j is non zero if i is equal to 0 and is equal to 0 otherwise, the dot product of other rows is 0. Furthermore we want that the rotation matrix is orthonormal, what this implies is that the dot product of a row with itself is 1 for example the rotation matrix which we considered. For the second example where the source was Lenna image which is an orthonormal matrix and now, another consideration to choose rotation matrix is to conserve energy. Why this is very important can be explained with the help of an example.

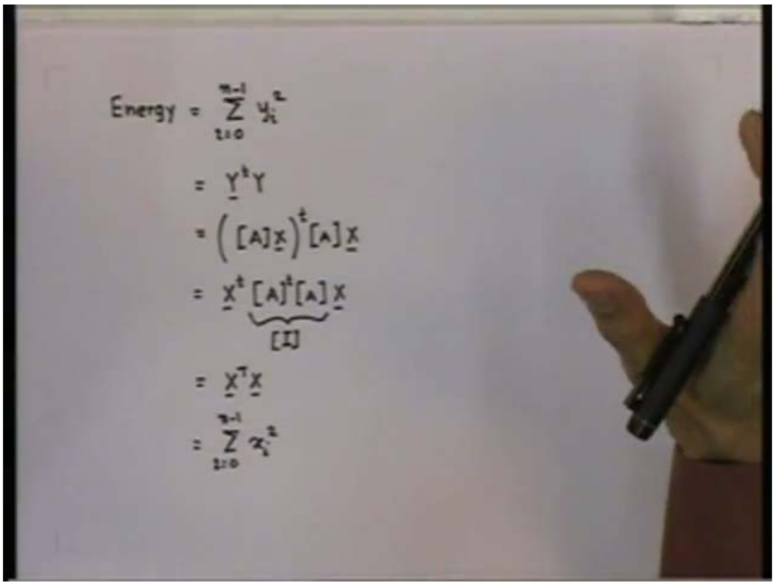
(Refer Slide Time: 34:39)



Let us choose an orthogonal matrix a given here and let us assume that we have the source output vector given by this. Now, the transformation A will give the transformed coefficients, coefficients as shown below. Now, if you look at the energy before rotation that is squares of this and addition will give you 81. Whereas, energy after rotation that is the sum of the squares of this elements will give you 324.

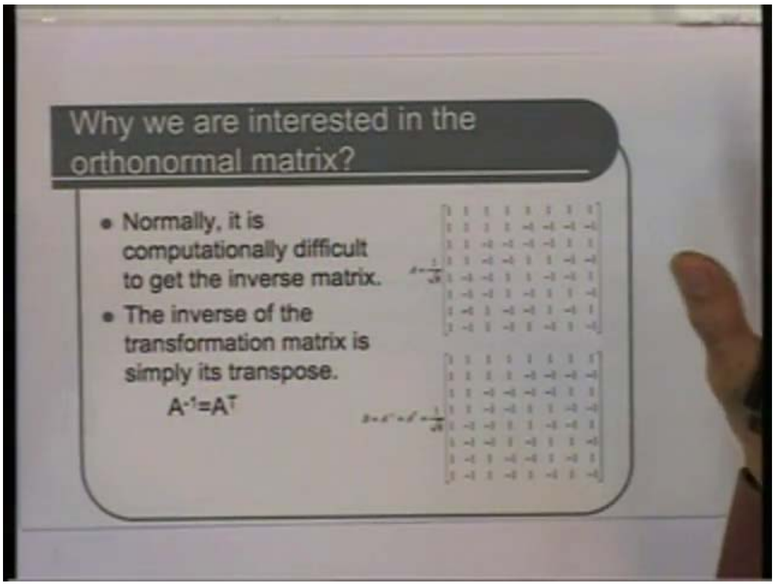
So, we do not want such a thing to happen we want that energy after transformation is also conserved, so this can be very easily done by scaling the orthogonal matrix by a scale factor. Now, the scaling does not change the fact that most of the energy is concentrated at the low frequency components, now we can formally prove this energy conservation as follows.

(Refer Slide Time: 36:02)



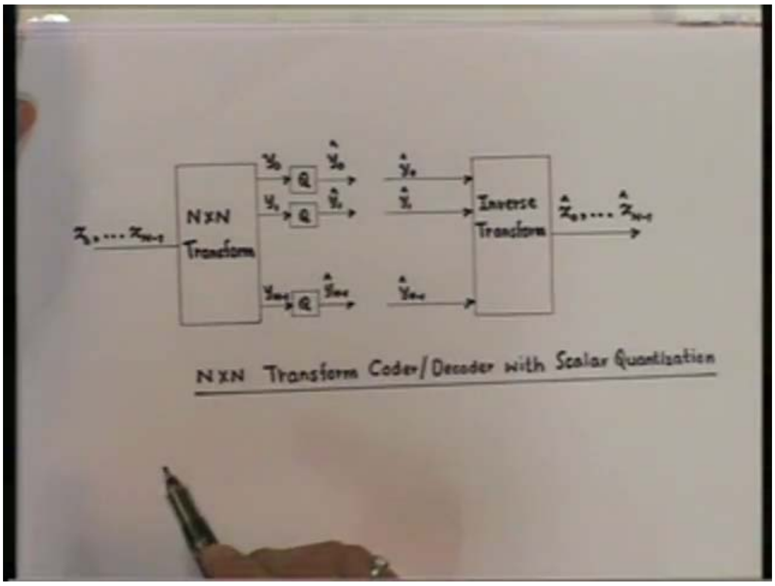
Energy in the transformed coefficients is given by y i squared i equal to 0 to n minus 1, which can be written in the form of matrix as dot product of y itself and this can be rewritten as which can be simplified as follows. Now, if we choose to be ortho-normal matrix then this is equal to x transpose x. Dot product of x with itself which is equal to the energy in the source output vector, so we choose the linear transform to be orthonormal matrix because it preserves the energy. Now, another reason for choosing this orthonormal matrix is explained in this slide.

(Refer Slide Time: 37:30)



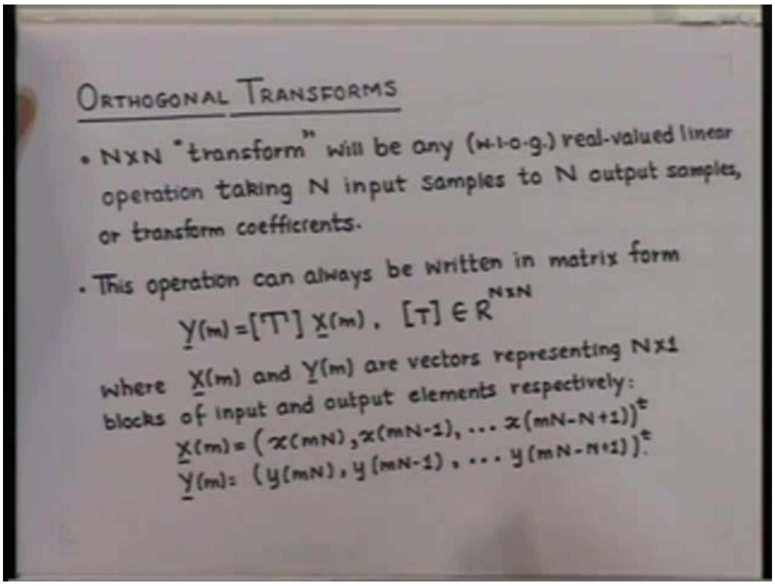
So, if we use a linear transform the in principle its inverse can be obtained, but it is computationally difficult to get the inverse matrix, but if this transform matrix is ortho-normal then the inverse of the transposition matrix is simply its transpose. Therefore, this has an advantage from this implementation point of view because the decoder is also a linear transform and mathematically it can be shown. That orthogonal transform will provide the least minimum squares error as far as the reconstruction is concerned; we throw off some of the transform coefficients. Now, we will not do this, but we will assume this fact and go ahead, so whatever we have discussed so far can be summarized in this block diagram here.

(Refer Slide Time: 38:47)



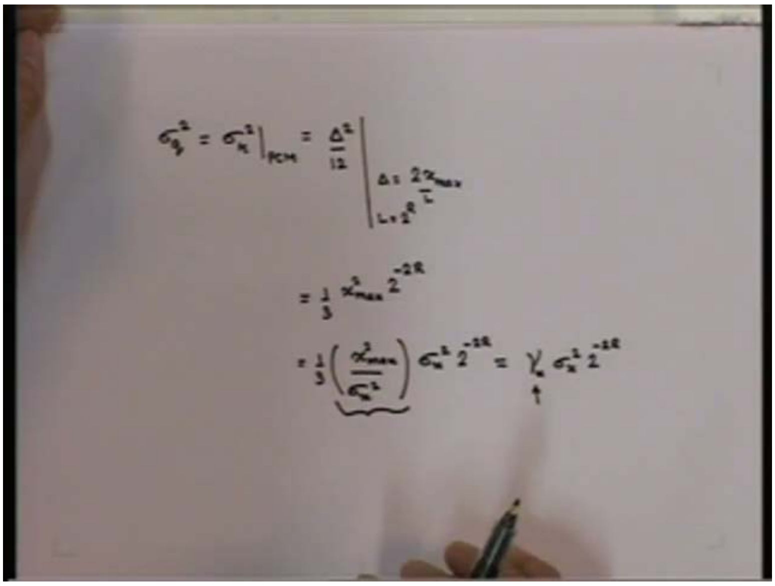
So, in transformed coding blocks of n input samples are transformed to n transformed coefficients which are then quantized individually and they are either transmitted or stored at the decoder. An inverse transform is applied to the quantized coefficients yielding a reconstruction of the original sequence. Now, by designing individual quantizers in accordance with the statistics of their inputs, it is possible to allocate bits in a more optimal manner, example encoding the more important coefficients at a higher bit rate. Now, for our discussion n by n transform will be any without loss of generality real values linear operation taking n input samples to n output samples or transform coefficients this operation can be represented in matrix form as follows.

(Refer Slide Time: 40:15)



Where x m vector and y m vectors represent n by 1 blocks of input and output elements respectively, where x m vector is related to the input sequence as follows and the y m vector is related to the output vector as follows. Now, before we go ahead with the discussion of transform coding, let us recall the quantization error which occurs in pulse code modulation. Pulse code modulation is nothing but discretization of analog wave form in time or space and amplitude.

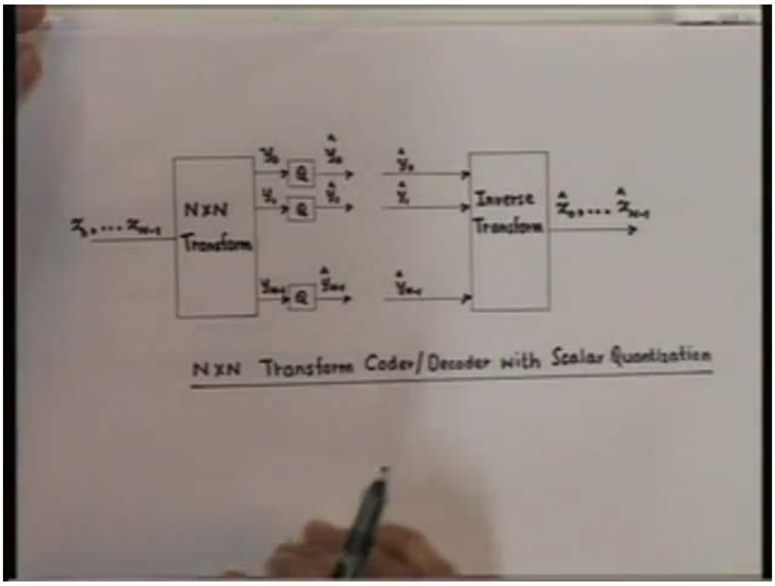
(Refer Slide Time: 41:33)



So, for a given bit rate of r bits per sample uniformly quantized P C M implies a mean squares quantization error of sigma q squared. This is equivalent to reconstruction error in P C M is equal to delta squared by 12, where delta is obtained by 2 x max divided by L where L is equal to 2 raise to R. This we have seen earlier, so this can be simplified as 1 by 3 x squared max 2 raise to minus 2 R, and this can be rewritten as x squared max by variance of the input multiplied by variance of the input and 2 raise to minus 2 R.

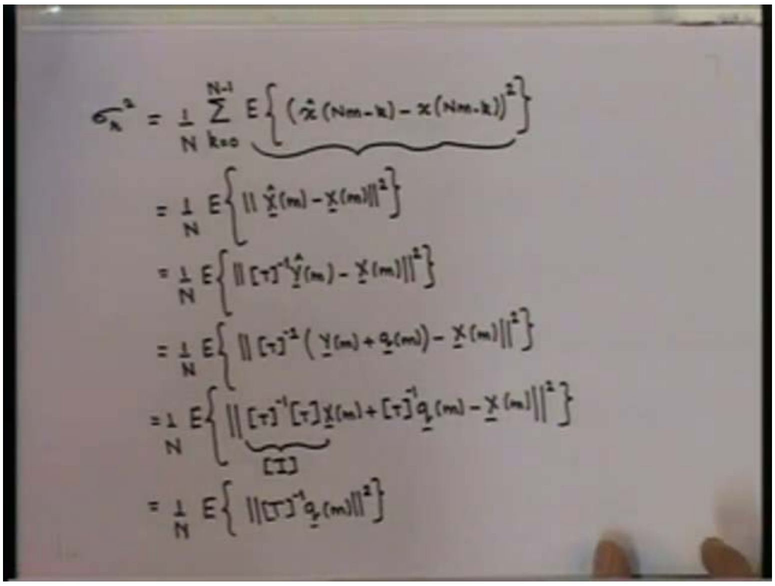
Now, this quantity depends on the distribution of x, so for a uniform quantizer for a given distribution, we can write the quantization error or reconstruction error as gamma x times variance of the input times 2 raise to minus 2 R. So, this constant gamma x depends on the distribution of x, we will utilize this information to find out the optimum bit allocation strategy in transform coding.

(Refer Slide Time: 43:47)



Now, another question that arises using this strategy what is the mean squared reconstruction error? That means error between the reconstructed sequence, and the original sequence.

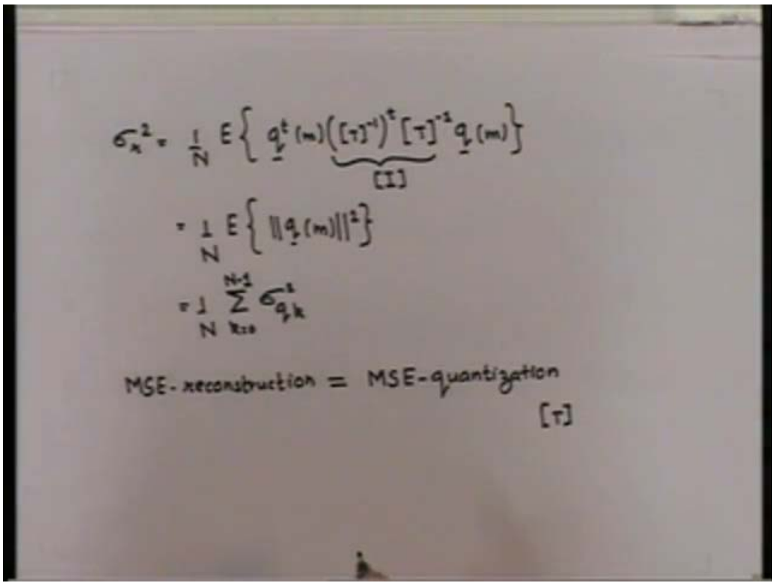
(Refer Slide Time: 44:04)



So, this can be written as mean squared reconstruction error is equal to 1 by N summation of k equal to 0 to N minus 1. Expectation of this is the mean squared error for a particular component and since there are n components, we get by definition this expression. Now, this can be simplified as 1 by N expectation of the Euclidian distance between the original vector and the reconstructed vector.

This can be rewritten as the quantized transform coefficient vector, which is equal to the original transform coefficient vector plus quantization error associated with each component. So, this is another vector minus x m and this can be further simplified as because y m is equal to transform times X m plus. Now, for orthonormal transform this quantity is equal to I and therefore, this reduces to 1 by N expectation of the inverse transform of the quantization error which can be rewritten as…

(Refer Slide Time: 48:13)



Now, because T is an ortho-normal transform this is again equal to identity matrix, so this is equal to one time and equal to. So, what this result indicates is that mean squared reconstruction error will be denoted as M S E reconstruction is equal to mean squared quantization error which will be indicated as M S E quantization. This is true when you have T as an ortho-normal transformation matrix, now in the example which we discussed earlier for lossy transformed coding application.

We threw off all the second components of the transform vector and quantized the first components of the transform vector. Now, the natural question is that assuming that transform is an N by N ortho-normal matrix. What is the mean squared error optimal bit rate allocation strategy? Assuming independent uniform quantization of the transform output. In other words we want to find out the rates for the different quantizer such that it minimize the average reconstruction error for the fixed given average rate. The next question is how do we choose this transform itself so that the mean squared error reconstruction is minimized. We will try to answer these questions in the next class.