Digital Voice and Picture Communication

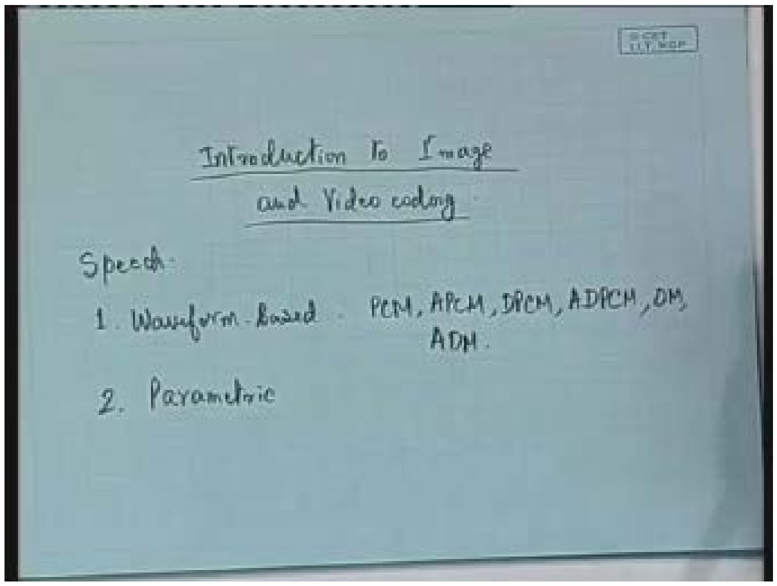
Prof. S. Sengupta
Department of Electronics and Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 16

Introduction to Image and Video Coding

An introduction to image and video coding: We have already given you a background of the speech coding and there you have seen that for speech coding we have followed two different categories of approaches. One is which was waveform based which included techniques like the starting with the PCM, adaptive PCM; then the predictive coding included the differential PCM, ADPCM, delta modulator, adaptive delta modulator etc. And we had yet another approach which we can call as the parametric approach.

(Refer Slide Time: 1:59)



Introduction to Image and Video coding

Speech.

1. Waveform-based - PCM, APCM, DPCM, ADPCM, DM, ADM.

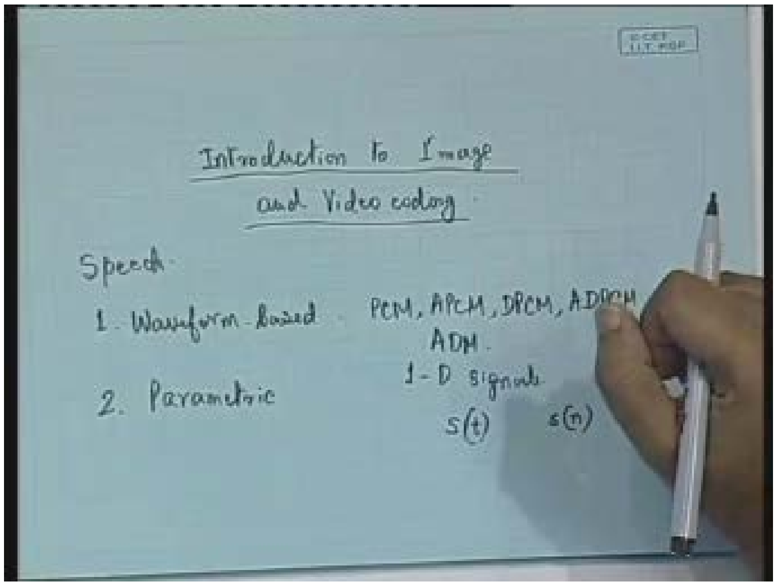
2. Parametric

In the parametric category we can place the LPC analysis, where essentially instead of encoding the speech waveform what we did was to estimate the predictor coefficients on a frame by frame basis and send the extra information like the pitch period estimation and all these things. So everything is in the parametric form and we had seen that parametric form had really helped us.

And another reason why the parametric form is very popular in speech is because; in order to have a speech synthesis we have a speech generation model. The actual speech generation process is modeled and we are making use of that so that really makes the speech coder quite efficient.

Now, in the waveform based again when we had analyzed this we made use of the one dimensional signals and what is the variability? It was varied with respect to time. So we had the signal which we were calling the original............. analog signal was s of t but in the discrete space we were calling the signal as s of n; the speech segment we were calling as s of n. This is what our input to the system was so it is a one dimensional signal.

(Refer Slide Time: 3:39)


Introduction to Image and Video coding

Speech

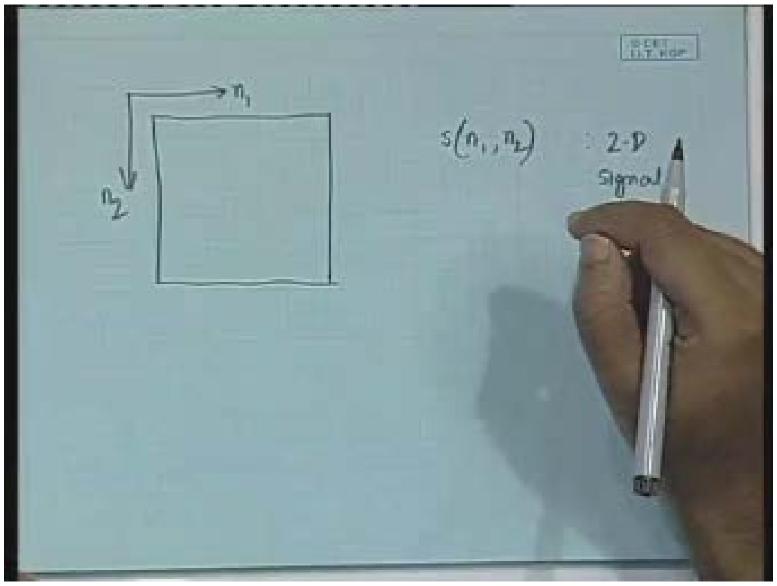
1. Waveform based PCM, APCM, DPCM, ADPCM, ADM

1-D signals $s(t)$ $s(n)$

2. Parametric

In the case of image and video the encoding philosophy is substantially different. First of all like the speech generation process we do not have any specific parametric model to be considered that is why the waveform based coding happens to be more popular for the image and video coding. But the first difference obviously comes in matter of dimensionality. You see that for the images for example, if it is if it is just a still image in that case how are we going to represent the waveform that waveform is essentially a two dimensional waveform. Because if we are taking an image which is an array of which is a 2 D array of pixels then the signal is represented as s (n 1 comma n 2) where n 1 and n 2 they will be representing the pixel indices. So we may be using .... this n 1 could be in this direction that means to say that in the x direction and in the y direction we can call this as n 2 so that the signal will be represented as s (n 1 n 2) so essentially it is a 2 D signal.

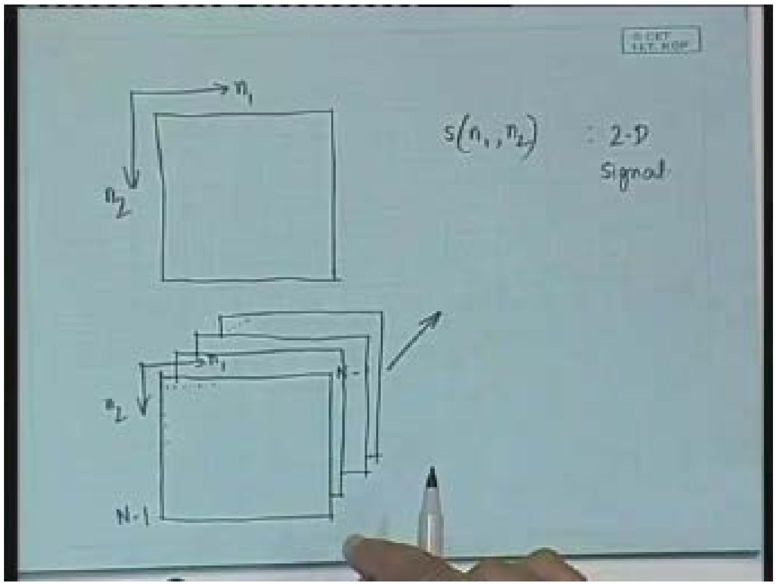
(Refer Slide Time: 5:13)



Now the variability in this case is not with respect to time. But the variability is with respect to space. So there are two parameters n 1 and n 2 which are essentially spatial index. So here, the sampling that is done for the analog image waveform, it is as if to say that we have an analog image waveform but whenever we are digitally representing it that analog waveform that analog 2 D waveform is sampled at some specific position at some specific positions and then we are representing this n (s n 1 n 2) as such sampled versions of the analog signal. So this is just in the sampled form that we are representing and it is a 2 D signal with variability in space.

In the case of video what we are having is a sequence of such images. So there what happens is that we are having one frame; now individually in the frame we are going to have the variability in n 1 and n 2 directions. So it will be sampled like n 1 is equal to 0 1 2 3 4 up to capital N minus 1 and here in n 2 direction also 0 1 2 3 4 up to capital N minus 1 that means to say that the image size is going to be an N by N image size and that composes a frame. But in the case of video we will be having a sequence of such frames.

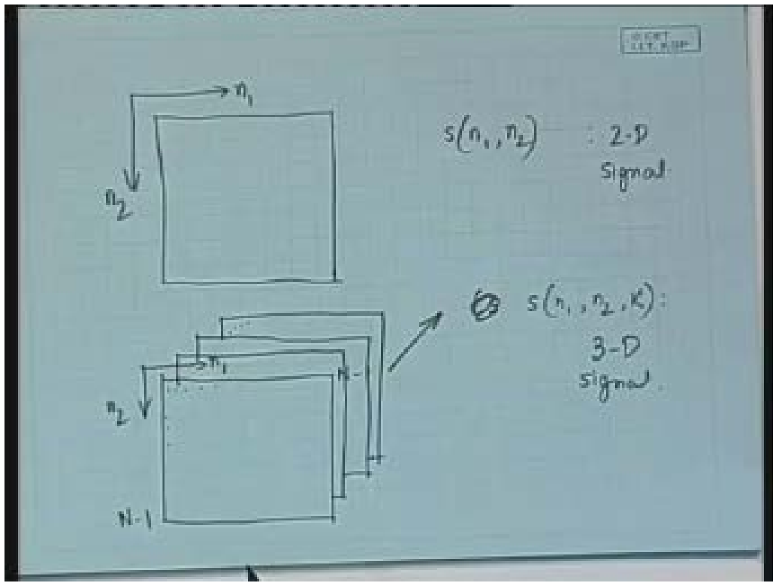
(Refer Slide Time: 7:09)



We also have a third dimension which is getting added to and that third dimension is going to be what? The variability with respect to time. So let us call in the continuous domain we would be calling it as the variable t but again because it is not continuous we have to make it in a discrete space so let us call this as s (n 1, n 2 comma k) where k basically will indicate what is called as the frame number.

So again time is also discretized because we are not continuously representing the video waveform but the video waveform is actually sampled at some snapshots. So every snapshot is a frame; every snapshot of the continuous waveform is what we are calling as a frame in the video sequence. So it is essentially treated as a three dimensional signal.

(Refer Slide Time: 8:22)



Hence, what it leads to is that there the fundamental bit rate what one obtains in the case of image and video is going to be much higher as compared to that of the speech signal. Because, after all y was a speech signal fundamentally not very challenging from an encoding point of view, you see. What was it? 10 kHz of sampling rate; you take, be your signal was restricted to at the most 3.4 kHz or 4 kHz for speech, 10 kHz sampling is the sample rate, have 8 bits per sample and then you are going to have a bit rate of 64 kilobits per second. But in this case you take the image of a typical size; let us take size of an image to be 1024 by 1024 so 1 K by 1 K which means to say that there is 1 megapixel. So 1 megapixel if we take and if it is a still image in that case 1 mega pixel and if we are having 24 bits per pixel, because for color images we are going to have as the R, G and B components so 24 bits per pixel in color which means to say that 24 megabits in one still image and just imagine that if this image changes at a rate of say 30 frames per second; if in a video we are having 30 such frames per second then the basic bit rate is going to be 30 times this so definitely that leads to an explosion in information and we have a requirement that the image and video signals must be compressed to a very significant extent, more so in video because in video the bit rate is getting much higher and not only that it has to be sent in real time. When we are saying about a frame rate of 30 frames per second in that case the entire processing or entire encoding process should be completed within one frame time. So we have only 30 milliseconds at our disposal and within that the entire encoding process needs to be completed so from that point of view it is highly challenging.

Now, on the other hand, as I was telling you that we are not having any definite parametric form that we can make use of, so we have to make use of the waveform based analysis itself.

Now what does a and an image encoder.................. so first of all that instead of dealing with both together: image and video, we will first be taking about the image encoding aspect and once we complete that we will go over to the video encoding part.

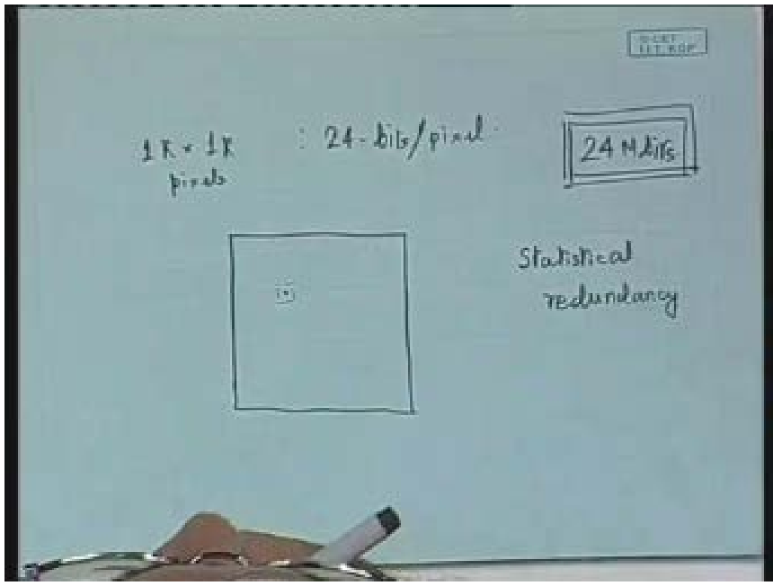
Now in image encoding what is going to be our total requirement. So, in order to transmit the image, first of all that whenever we are having the images there happens to be............. now although I was mentioning that the fundamental bit rate is excessively high that is say to that for a 1 k by 1 k 1 k by 1 k image having 24 bits per pixel 24 bits per pixel is and 1 k by 1 k is a number of pixels in the image, we are going to have 24 megabits as the total number of bits that we will be requiring for this entire image to be transmitted which is a very high number. But very luckily what happens is that the redundancy which is present in all most all types of natural images is very high; redundancy in the sense that if you are taking an image take a pixel anywhere in the image, you will be finding that this pixel is highly correlated in its intensity value with all its neighbors.

So if you are finding out its neighbors, may not be the immediate neighbors alone but which are quite close by, that means to say that may be that 4 or 5 pixels on this side 4or 5 pixels on the other side up and down so if we are taking this entire neighborhood we will be observing a significant amount of correlation. So we have to basically utilize this amount of redundancy in our encoding process.

Now what is that redundancy?

Because after all image when we are representing it is represented as an array of numbers. So there is redundancy in the pixel intensity values or rather to say the numbers with which we are representing the pixel intensities. So this in a so this sort of a redundancy which is evident out of these numbers, this will be referred to as what is called as the statistical redundancy.

(Refer Slide Time: 14:22)



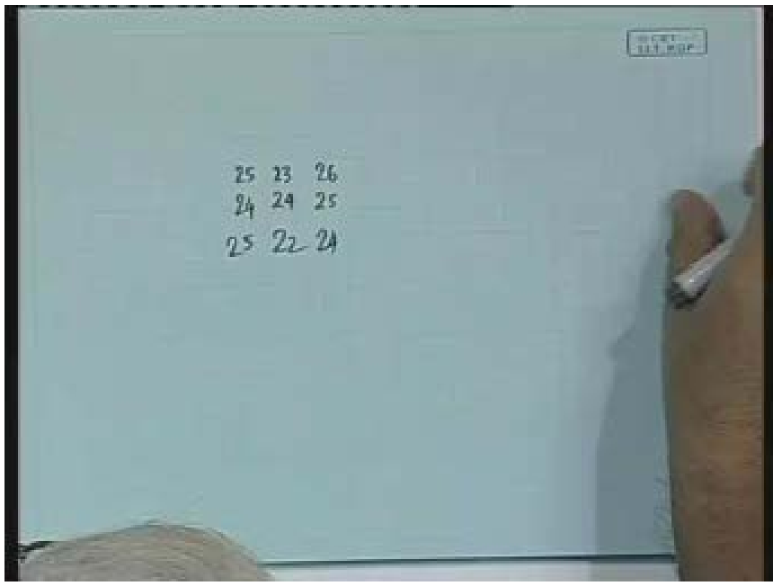
Statistical redundancy essentially tells us about what is the redundancy that is existing in space in between the samples or the image samples which are in the close neighborhood. Hence, this gives rise to what is called as statistical redundancy.

Now there is yet another redundancy which is exploited in the image coding and that is to say it in a simple layman language, how to fool our eyes. You see, ultimate detector of the image or video in any image or video processing system is the human eye. The human eye is going to tell based on the perception that we are going to have; perception of the image or perception of the video sequence what we are getting through using the eye as our sensor there we have to see that to what extent our eyes can distinguish the different aspects of the image or video signal and we should be restricting our accuracy or restricting our representation to only what our eyes can see.

So what our eyes cannot perceive we can rather avoid those kind of information. So there is yet another redundancy which we have to exploit and that we are calling as psychovisual redundancy.

Psychovisual redundancy essentially refers to what our eye does not really require. So using the psychovisual model lots of studies have been made. Now, how the two redundancies are made use of? To make use of the statistical redundancy it is definite that we have to do some kind of a number jugglery. Number jugglery means that wherever the numbers are quite similar if we have a patch of images; supposing we have a patch of image where say the pixel intensity values happen to be like this: 24 25 23 24 25 22 24 25 like this so you can see that here the average intensity may be of the order of 23 or 24 and there is a small amount of variation which is there. So we can have a DC representation which will say that the average intensity of this is say 23 or 24 whatever it comes to and then we can also decide to give some kind of an incremental representation that, over and above this DC value the individual pixels are going to have some variability pattern as something like this that where the variability could be the just the numbers which are the differences with the immediate past values which obviously leads to a differential encoding.

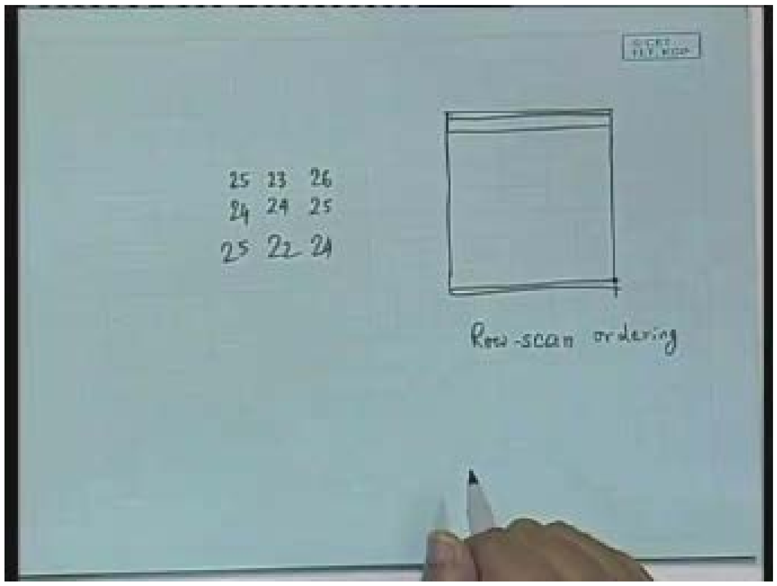
(Refer Slide Time: 18:07)



25	23	26
24	24	25
25	22	24

So here, if we are sending the first pixel let us say first pixel is 25 and then the next pixel we are sending as 23, next as 26, next as 24 actually there is a definite way in which the image gets scanned and that you must be knowing that it gets scanned from the top left to bottom right. So top left it covers the first row of the pixel array then it goes to the next row of the pixel array and so on and at the end it goes up to the last row of the pixel array and the last pixel to be picked up is the last row last column pixel. So this is the normal row scan ordering, row scan ordering. It is ordered row-wise. First complete a row then go over to the next row this is the order in which it goes on.

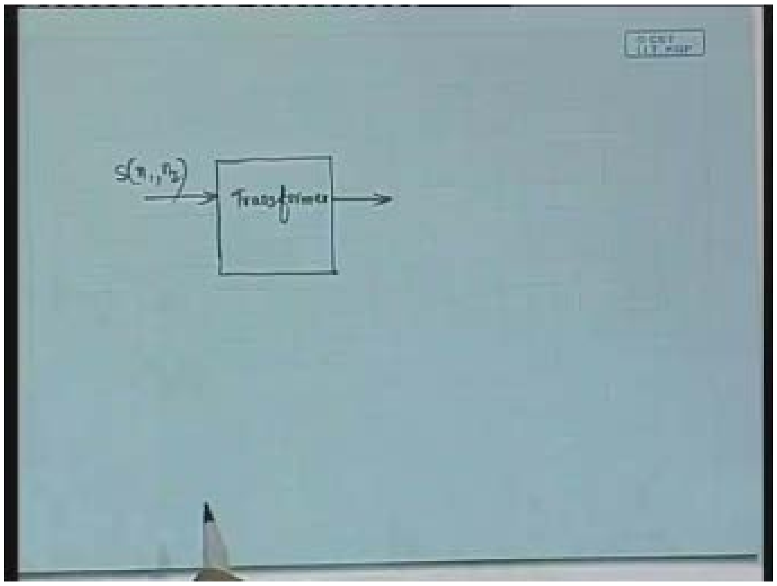
(Refer Slide Time: 18:54)



That is why you can say that when the pixels are scanned like this it is as if to say that a sequence of one dimensional pixels will be available to us and using those pixels it should be possible for us to exploit the redundancy in the sense that there is definitely something that the differential coding is going to help us.

Actually in a typical image coding system what we are going to have is like this that as a generic block diagram Forman was presenting, the images will be first put through a process of transformation, I will call this generic block as the transformer which will be some kind of a transformation that this image array will go through. So s(n 1, n 2) happens to be the input to the system and this transformer is going to have some kind of transformation and what kind of transformation; a transformation that makes the image signal amenable to good compression. So there should be a good amount of compression that we should be able to achieve so the transformation should be designed accordingly.

(Refer Slide Time: 20:32)

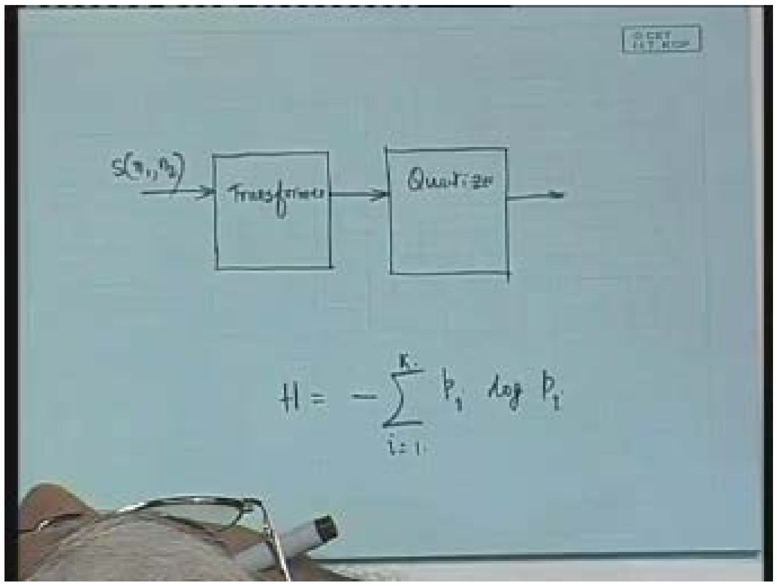


Now, after it is transformed, this transformed image array what we get, that one should be able to quantize, because if we want to represent all the transformed values without going through any quantization process, the amount of information that will be required will be excessively high, that is why we need a kind of a quantizer which should follow the transformer. And this quantized output In fact this quantizer should be so designed in order to exploit the psychovisual redundancy and this transformer should be so designed to make use of the statistical redundancy.

Thus, transformer exploits the statistical redundancy, quantizer exploits the psychovisual redundancy but when you give quantizer as a block, please remember that quantizer essentially leads to a loss in the signal. the loss in the signal means that you are truncating some values so no exact reconstruction of the signal should be possible at the receiver end because at the receiver end what we have do is to just reverse this entire process.

Now these quantized signals also seem to have some amount of redundancy and that is also a form of statistical redundancy. Actually what happens is that there we go to the aspects of information theory. In information theory what it tells us is that the symbols......... I mean, when we have to transmits something and we have a set of symbols to be transmitted in that case the symbols which are more probable they should be transmitted with less number of bits and the symbols which are less probable they should be transmitted with more number of bits so that by having the policy that more frequent symbols requiring less number of bits would obviously lead to the fact that you will be saving the bit rate rather than assigning equal number of bits to all the symbol, so essentially there is going to be some exploitation of the entropy of the system.

(Refer Slide Time: 23:57)

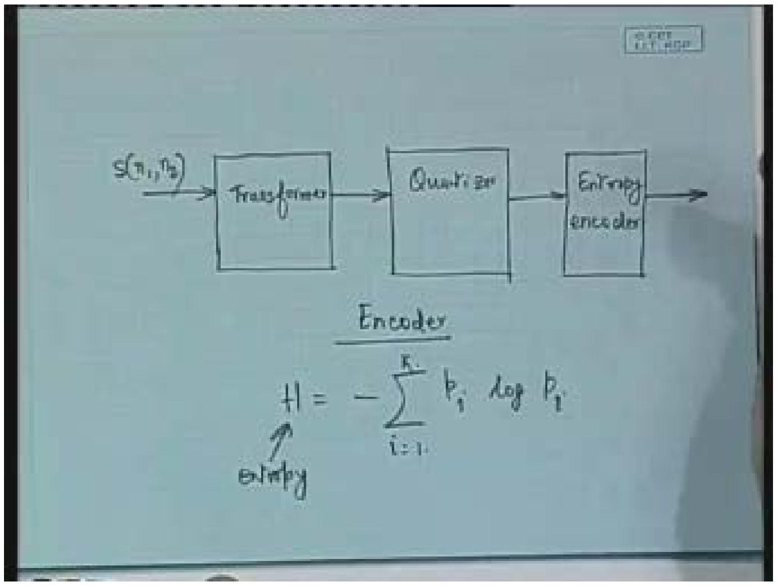


Every symbol at the quantizer output will be associated with some probability of occurrence and based on the probability of occurrence we can compute the entropy.

What is the entropy expression?

Entropy expression is that the entropy is going to be H is equal to minus p i log of p i and this is going to be summed up for i is equal to 1 to k if you have to transmit k number of symbols. So this H is called as the entropy and you all will be knowing Shannon's entropy coding theorem for lossless channels which says that the minimum attainable bit rate that a system can have is going to be its entropy so you cannot really compress beyond its entropy value that is what Shannon's theorem on noiseless channels is going to tell us. So we must be having some kind of an entropy coding in order to encode the quantized symbols. So we must follow it up by an entropy encoder entropy encoder and these coded signals these coded bits will be put into the channel for transmission.

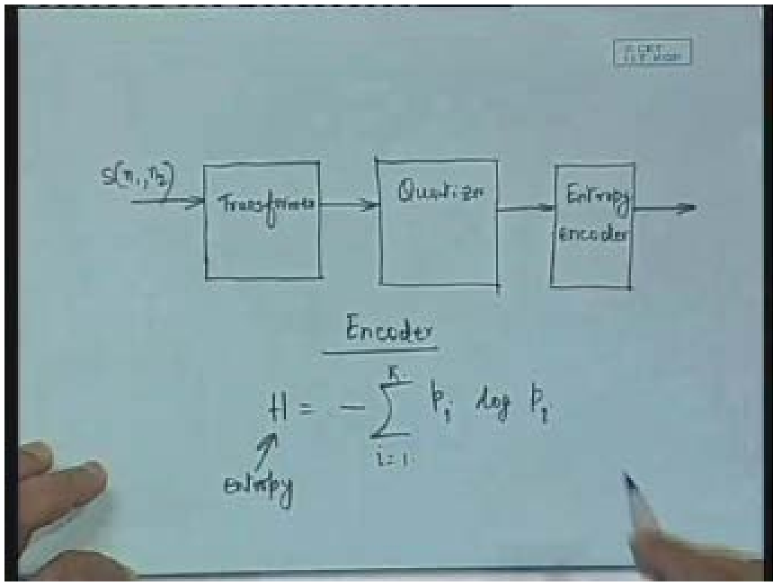
(Refer Slide Time: 25:21)



Now we have to do a reverse of this process at the decoder end. This is our encoder end and at the decoder end we must do the exact reversal of this. so what we have to do, I need not have to draw the block diagram, it will be for the entropy decoder, it will be a dequantizer, it will be an inverse transformer and at the end what will you get? s cap of n 1, n 2 so s cap of n 1, n 2.

Just to quickly show you, this is the entropy decoder (Refer Slide Time: 25:47), this is the dequantizer and the next one is going to be inverse transformer and the output is going to be s cap of n 1, n 2; so this is going to be our decoder.

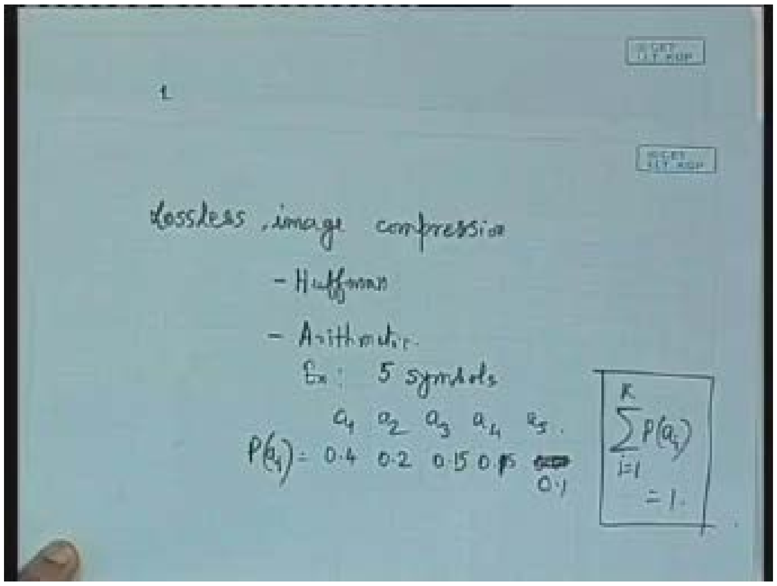
(Refer Slide Time: 26:11)



Now, in this process, this entropy encoding process is a totally lossless process. So this block is a lossless block (Refer Slide Time: 26:23), quantizer is a lossy block and transformer is yet another lossless block. Now somehow if we could encode without using the quantizer in that case what would have resulted is a lossless compression. Lossless compression is possible to achieve, why? Because whenever we are getting the samples, whenever we are getting the image samples even if we do not quantize this samples still we can reduce the number of bits, just the first one and then rest if you are encoding by the differential pulse code modulation in that case what you are required to send is only the incremental information which is going to take much less bits as compared to encoding individual pixels because the entropy of the differential signal is much less as compared to the entropy of the original signal. So it will be requiring much less number of bits to encode and we will be able to achieve compression and that compression is lossless because we need not have to truncate those values; even if we transmit those values without truncation still it is going to give us a good amount of compression. But there is a limit up to which the compression can be achieved and that is given by Shannon's coding theorem.

But given the fact that image or video both these domains have very high amount of information content, the amount of compression that we will be achieving by simply going through a predictive coding followed by the entropy coding is not going to be very sufficient for us. We must be going in for some kind of a lossy compression. There are two compression methodologies: one is what is called as the lossy as the lossless image compression schemes lossless image compression and in the category of lossless image compression one can list out all the entropy coding techniques; and what are those? There is Huffman coding technique which most of you must be knowing because what does Huffman coding do? Huffman coding; it prepares a table of coded values which is based on the probability of occurrence of the individual symbols. So based on that the Huffman coding table will be prepared and we are going to give the so based on that coding table we will be encoding the individual symbols and achieve a good amount of compression in the process.

(Refer Slide Time: 31:56)



lossless image compression

- Huffman
- Arithmetic

Ex: 5 symbols

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5$.

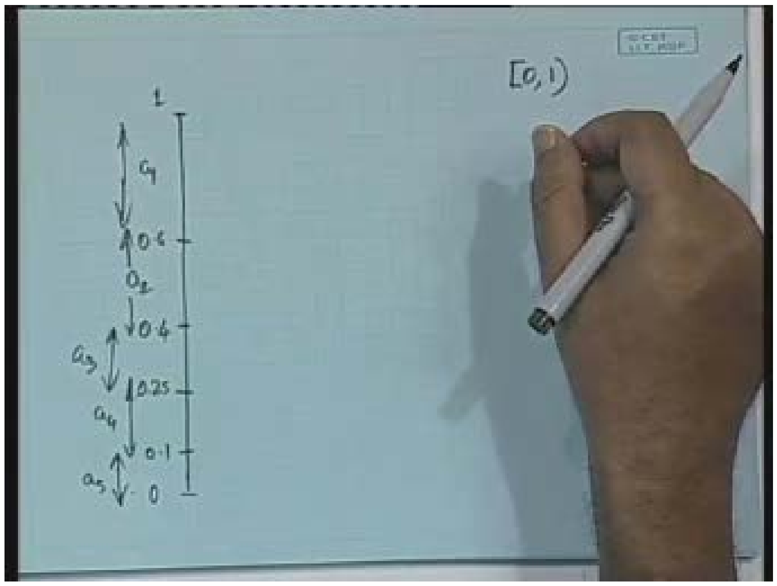
$P(a_i)$: 0.4 0.2 0.15 0.15 0.1

$$\boxed{\sum_{i=1}^{K} P(a_i) = 1}$$

There is yet another popular lossless image compression technique and that is called as the arithmetic coding. This is also a form of lossless encoding, only thing is that here in the case of arithmetic coding the symbols which exist they are represented in the form of a number which is in the range of 0 to 1. So, arithmetic coding goes like this. Many of you also must be knowing what arithmetic coding is; that supposing we have got say five symbols, so say as an example we say that there are five symbols and five symbols are a 1 a 2 a 3 a 4 a 5 and let us say that a 1 is having a probability of 0.4; a 2 is having a probability of 0.2; a 3 having a probability of 0.1, this has 0.05, this has 0.05 does it add up to 1? 0.6 0.7? No, the 0.15, 0.15 so 0.3................ 5.9 and this has 0.1 so if these are the probability values; p of a i values are like this so we have the sum total of the p i p(a i) values where i is equal to 1 to k that must be equal to 1 the sum total of the probability so this is leading to 1 so we have all these probability values which we are going to represent in the form of a number line.

So let us have a number line and a number line is going to us something like this; say here we represent 0 and here we represent 1 (Refer Slide Time: 31:44). Now we subdivide this number line into five parts and those five parts will be proportional to its the length of those five parts will be proportional to their probability values (Refer Slide Time: 32:00). So 0.4 is for a 1 so we cut out this part say 40 percent of the length we just cut out so let us say that in this case if we take 0.6 so between 0.6 to 1 this we can call as a 1 so a 1 will be represented by this subpart of the line (Refer Slide Time: 32:26) the total range is (0 comma 1). So this is an half-open half-open interval of (0 to 1) so 0 is included but 1 is not included so it is without including this one so (0 to 1) so 0.6 to 1 this sub-range corresponds to a 1 and then the sub-range of 0.4 to 0.6 0.4 to 0.6 this sub-range corresponds to a 2 and then the sub-range of 0.25 to 0.4 that corresponds to a 3 and 0.1 to 0.25 this sub-range corresponds to a 4 and 0 to 0.1 this sub-range corresponds to a 5.

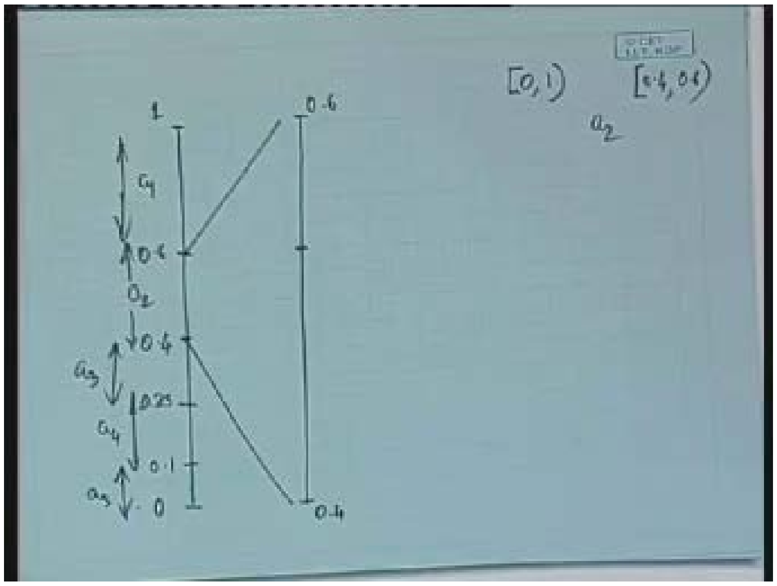
(Refer Slide Time: 33:34)



So we have partitioned this line into five such proportional divisions a 1 to a 5. Now what we have to do is that when we receive the symbols, let us say that we have received the first symbol as a 2 so in that case a 2 is where; between 0.4 and 0.6, so to encode this a 2 we know that it has to be represented by any number that falls within the range of 0.4 to 0.6. Therefore, the first symbol is a 2, so naturally we know that 0.4 to 0.6 is the range which we have to represent.

Now, in order to represent the symbols that follow after a 2, we must expand this segment of 0.4 to 0.6 to length that is same as that of this 0 to 1 range so we just expand this and now we represent another number line and in this case the lower range is 0.4 and the upper range is 0.6. So it is covering the half open interval of 0.4 and 0.6 end is the open one so it is now using [0.4 comma 0.6) half open. And now we have to subdivide this line [0.4 to 0.6) again into five such proportional parts.

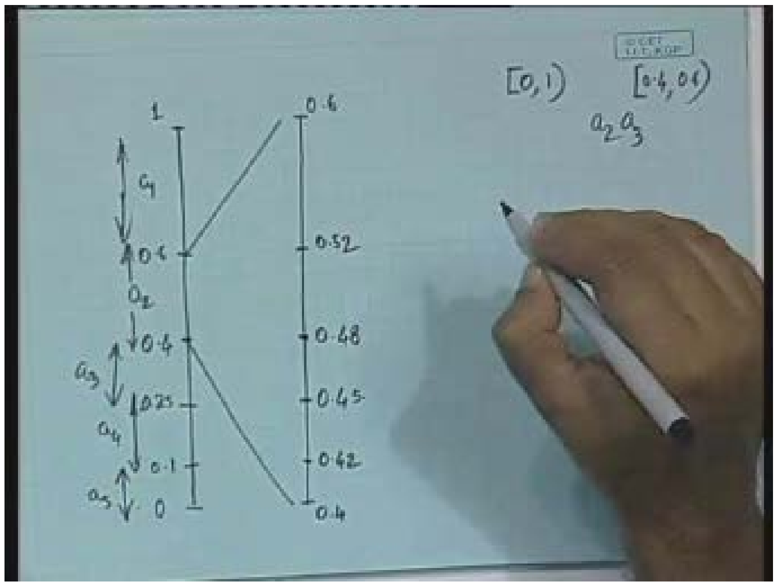
(Refer Slide Time: 35:45)



Therefore, this will be a 60 percent of this will compose the a 1 so what do we write for a 1? So 60 percent of this means that 0.4 to 0.6 so it is a total range of 0.2 so this will be this will be 60 percent of 0.2 60 of 0.2 means 0.12 so this will this will lead to 0.48 am I right? 0.52............... oh sorry yes [0.4 to 0.6 0.2 and 60 percent of 0.2 means 1.2; 1.2 plus, it is 0.12 so 0.4 plus 0.12 is 0.52 you are right.

So this is 0.52 and this is going to be point 0.48, this is going to be 0.48 (Refer Slide Time: 36:13) and what is going to be this one? This will be 0.4 to 0.6 0.2 so 0.2 25 percent that means to say that 0.05. So this will be 0.45 and this is going to be 0.42 yes, 0.42, so this will be the sub-parts.

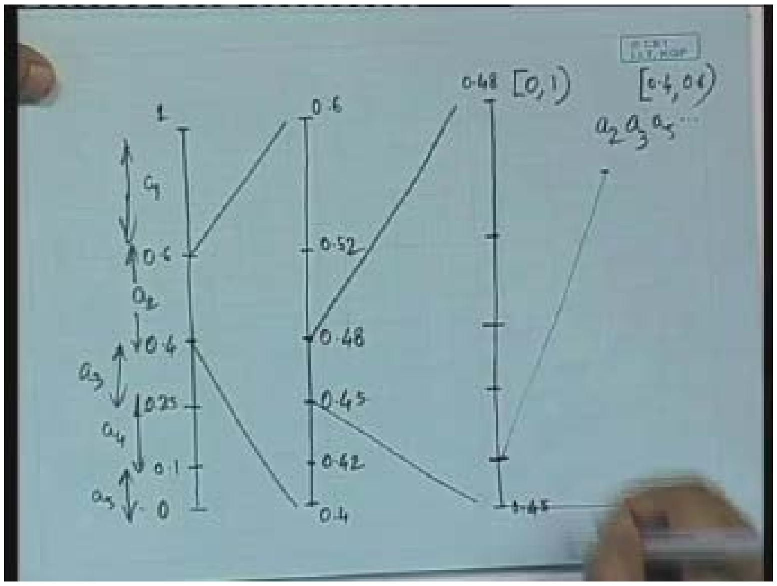
(Refer Slide Time: 36:47)



Thus now, if the next symbol in the sequence happens to be a 3 let us say, in that case what we have to do is that this a 3 has to be expanded. So now we are going to have the number line that goes from 0.45 to upper limit is 0.48. So now what is the conclusion? This subsequence what we have obtained a 2 a 3; a 2 a 3 is forming a sub-sequence of the entire sequence; this a 2 a 3 sub-sequence is getting represented by a number in the range of 0.45 to 0.48. But exactly where it will be? That will be decided by the next bit. So again this 0.45 to 0.48 we have to subdivide into five proportional parts as before and then we have to get the next symbol supposing the next symbol is a 5 so a 5 means now this has to be expanded so it will be 0.45 to some other number so it will be a 2 a 3 a 5 will be represented by some number in this range. Therefore, like this we proceed for the arithmetic coding. So ultimately the entire sequence of symbols will be represented by a number that is the basic philosophy of arithmetic coding.

Arithmetic coding when we compare that with Huffman coding the difference is that in the case of Huffman coding we have got some unique codes which are assigned to the individual symbols whereas in the case of arithmetic coding we do not have such one to one correspondence between the symbol and this code. Rather than doing that what it does is that it takes the sequence as a whole and tries to encode the sequence using a number using a real number in the range of 0 to 1 and ultimately it is this number which will be encoded into the bit stream. That is the basic philosophy of arithmetic coding.

(Refer Slide Time: 38:07)



Now it is seen that arithmetic coding achieves a greater accuracy. Arithmetic coding can lead to the average to the number of bits per symbol........... see, number of bits per symbol in the Huffman coding is always an integer because every symbol is going to have some number of bits, so it is either 1 bit for some of the symbols, 2 bits for some of the symbols, 3 bits for some of the symbol so it is always an integer whereas in this case it is going to be a real value; the number of bits per symbol that is going to be a real value.

So arithmetic coding has got advantages as compared to the Huffman coding and in fact it is seen that arithmetic coding leads to a greater coding efficiency or rather it approaches the value of the entropy much closer as compared to the Huffman coding approach. That is also another form of the lossless image compression.

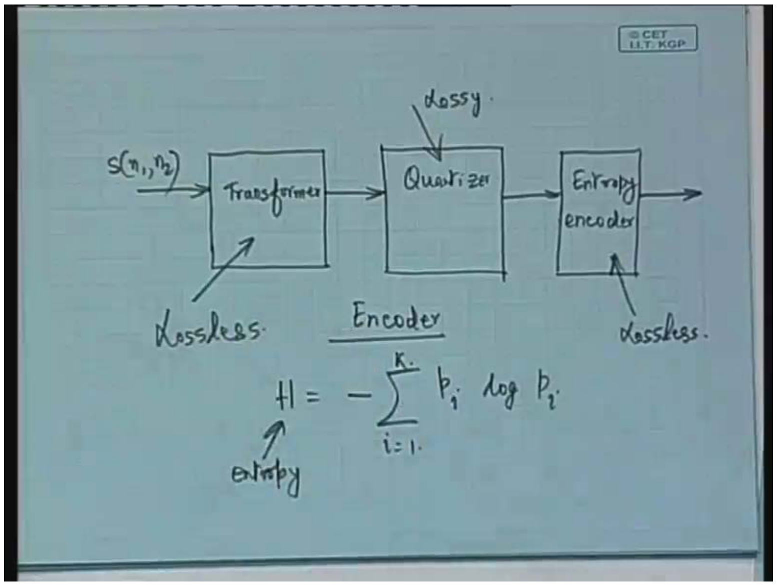
But as I was telling you, there are many other lossless compression techniques also available. We started with Huffman, then there is arithmetic coding and then there are other coding techniques like the Lempel-Ziv coding. Lempel-Ziv coding, as you know is that is that is yet another lossless coding technique which is used in let us say for the PDF the portable document file format, the PDF format that makes use of the Lempel-Ziv coding. Or even for the zip files, whenever we are compressing the files that make use of the Lempel-Ziv coding so that is also a lossless compression technique. So there are several such lossless image compression techniques which are in existence. But as I was telling you that lossless compression techniques always have got some limitation; limitations because that you can achieve only a limited amount of compression using the lossless techniques. And it is seen that even if you make use of a quantizer and you try to quantize the transformed outputs, in many of the cases our eyes will be unable to detect any major perceptual degradation in the quality of the image although number-wise it is a lossy image.

Because, if the s cap (n 1, n 2) array does not exactly match with s (n 1, n 2), arrange we are going to call that reconstruction as a lossy reconstruction. But visually if you do not see any perceptual degradation in s cap (n 1, n 2) then you may as well as accept it and if you are accepting some lossy image compression scheme in that case such kind of a lossy image compression schemes are going to be much more efficient.

So what is the lossy element?

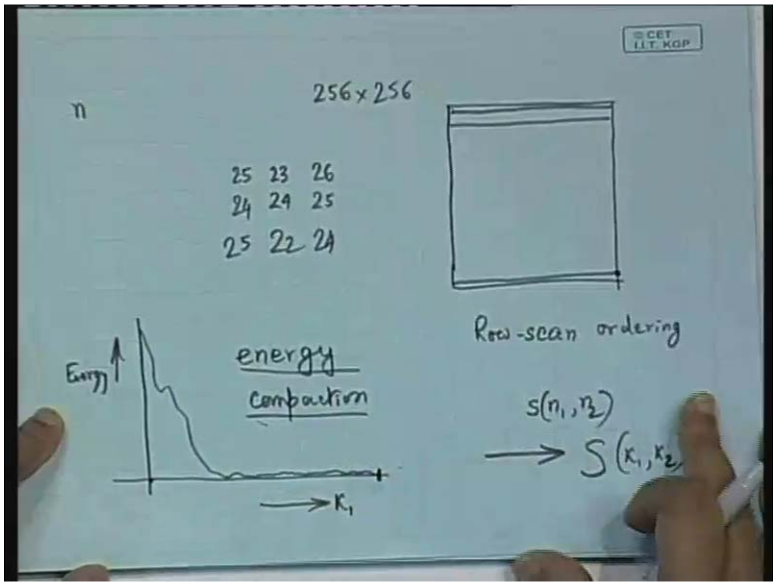
Lossy element is the quantizer about which we already know. But what must be understood is that, in order that we can efficiently achieve a lossy compression, the trick lies in the transformation block which we had mentioned.

(Refer Slide Time: 00:43:20)



So it is this transformation block which is essentially lossy which is essentially lossless but this block follows a very major role in the sense that it makes the signal amenable to compression. That means to say that it exploits the correlation that exists in the signal and essentially what it what the transformation process tends to do is what is called as energy packing. See, the concept of energy packing is something like this that; let us say that we take the example 3 by 3 array what we mentioned, this 25 23 26 24 24 25 25 like that.

(Refer Slide Time: 00:44:05)



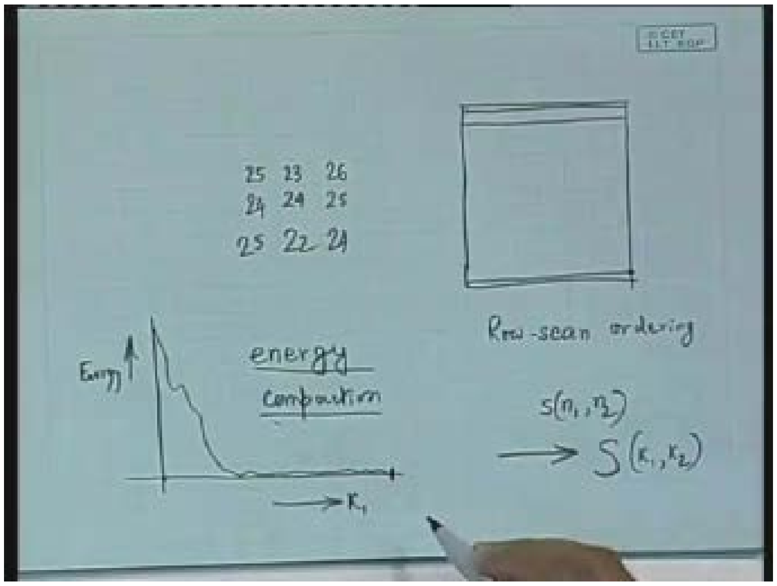
Now it is having this 3 by 3 image is having a very high average value. The average value of that is of the order of the 23 or 24 and then there are some variations, but those variations are less. So in terms of a frequency domain representation if we think of in that case in the frequency domain we will be saying that there is a small amount of variation that takes place in the frequency domain.

So, if we are plotting the energy of the signal versus the frequency; in this case frequency means the spatial frequency, in that case we will be finding that predominant amount of energy is there in the DC component itself and some low frequency portions of the signal spectrum will also contain some energy. But the high frequency components will be having very minimal amount of energy.

So if we obtain some kind of, not for this small example image but for a general image let us say that this is where I plot the frequency (Refer Slide Time: 45:32) so what we want to do is that from s (n 1, n 2) that is to say the original image array, we must transform that into a transformed space which we are indicating as S (k 1, k 2). We use the capital letter; in place of small s we will be using capital S in order to denote the transformed values and instead of using the indices as n 1 and n 2, in the transformed domain the corresponding indices will be presented as k 1 and k 2. So s(n 1, n 2) is being transformed into S (k 1, k 2). So let us say that we have a simple one dimensional plot in anyone direction; let us say in the k 1 direction; along the k 1 direction we take a plot about the total energy about the energy of the signal.

So in this case let us say that we represent the energy or the squared value of the pixel and then in the frequency domain if we plot the value, k 1 is equal to 0 corresponds to the DC component, so if we find that the DC value is quite high and if the spectrum is something like this (Refer Slide Time: 46:59): say this is the last k 1 value, in that case what is the conclusion that we can drop from this energy spectrum that there is what is called as energy compaction.

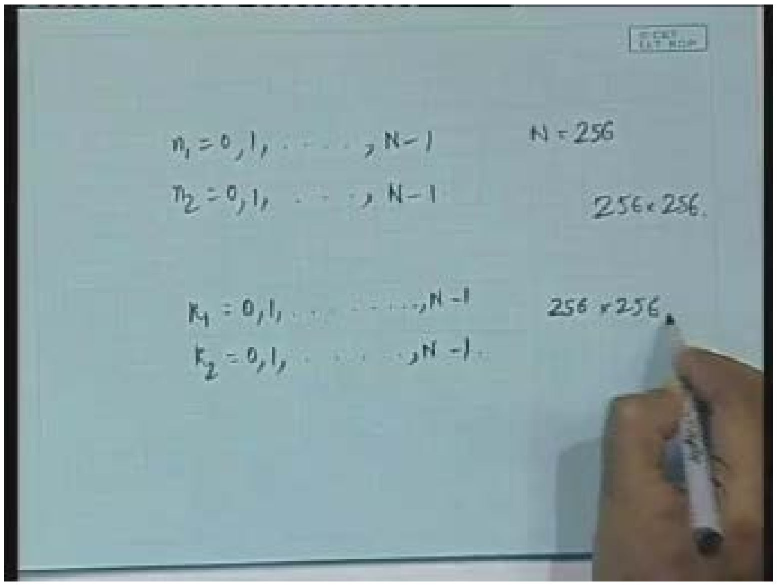
(Refer Slide Time: 47:19)



Hence, the transformer that we are going to use that should offer some energy compaction property. Energy compaction means using only a few spectral coefficients or only a few k 1 coefficients. We should be able to have most of the energy, like say for example, I give a typical example, let us say that the image size is 256 by 256 pixels, so n 1.......... actually to represent the image that way, we have n 1 as 0, 1,........... up to n minus 1 and n 2 as 0, 1,..................... up to n minus 1 and in this case we have used n is equal to 256 that is why we are saying that the array is

256 by 256; of course it is a square image example but if we have a rectangular example rectangular image example in that case this n and this n would not have been the same, we would then called n 1 as 0, 1,.............. up to n 1 minus 1 and n 2 as 0, 1,............. up to n 2 capital N 2 minus 1 and in that case the image size would have been a rectangular one capital N 1 by capital N 2 so you could have a size of 176 by 144 or you could be having 352 by 288 any such rectangular sizes should be possible.

(Refer Slide Time: 49:35)

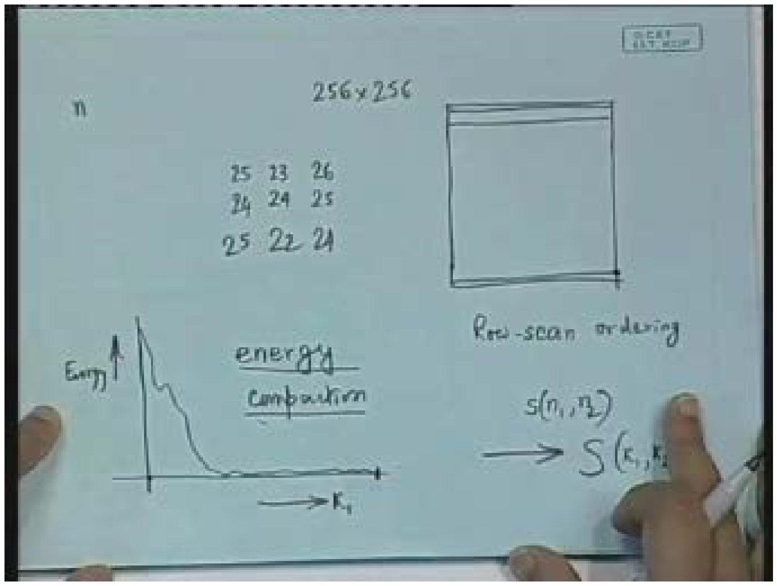


$n_1 = 0, 1, \ldots, N-1$
$N = 256$
$n_2 = 0, 1, \ldots, N-1$
256×256
$k_1 = 0, 1, \ldots, N-1$
256×256
$k_2 = 0, 1, \ldots, N-1$

When this is the image size and you are taking a transformation, all the unitary transformation that we can take leads to the fact that even in the transform domain indices whenever we are having k 1 and k 2 the transform domain indices also varies from 0 to N 1 minus 1 to N minus 2 and k 2 also 0 to N minus 1 which means to say that in the k 1 k 2 space also we are going to have 256 by 256 array. That means to say that we are going to have in S(k 1, k 2) array we are going to have 256 by 256; in other words, 64000 coefficients.

Now out of this 64000 coefficients if most of the coefficients does not carry energy with it, if only 10 percent of this array, say S (k 1, k 2), if only 10 percent of these coefficients carry 90 percent of the energy in that case we should be preserving the 10 percent of the coefficients and discarding the 90 percent of the other coefficients which are only carrying remaining 10 percent of the energy. In that case we will be able to achieve some significant amount of compression in the process. But to make it amenable to compression this kind of energy compaction was indeed necessary.

(Refer Slide Time: 50:45)

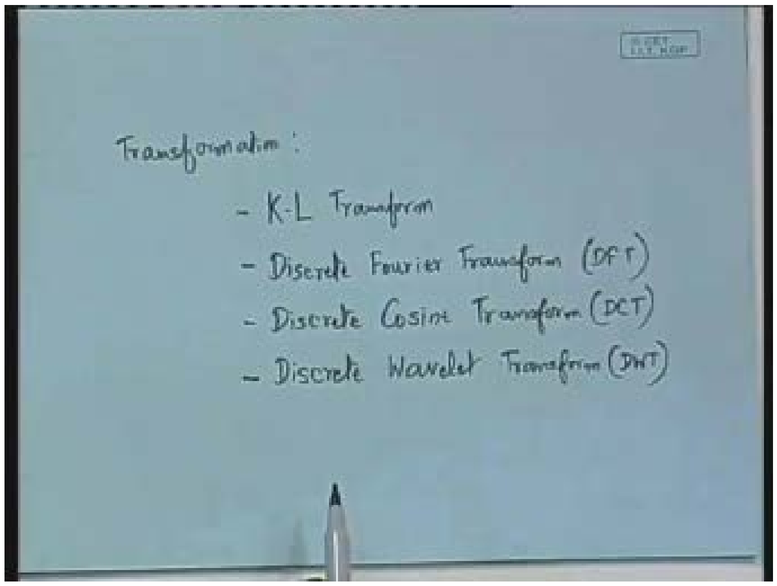


We could not get this picture in the original spatial domain. But we can get this picture after studying the redundancy that is present in the original picture domain and the redundancy is best depicted in the transformed space. So, while carrying out the transform space analysis we will be finding that there is an energy compaction and that energy compaction essentially leads to the fact that we can achieve good amount of compression if we first use the transformation followed by a quantization.

Now in the in so now there are many such transformation techniques which are which have been proposed till date. In fact there is a big amount of research that has gone into this process; I mean, finding out the transformation. In fact the transformation's essentially requirements are that it must be an energy preserving transformation that is the first requirement and second is that the transformation that we do must be an orthogonal transform which means to say that using the coefficients by which we are representing the image in the transformed space all these individual coefficients, they should be linearly independent of each other so that the transformed coefficients are effectively decorrelated. Because we want that the transformed coefficients are decorrelated from each other and that is something that we are doing by the process of the orthogonal transform.

Now in that category in the transformation there are several popular transformations that are in existence and in that we have first the optimal transform which is the Karhunen-Loeve transform so this referred to as the K-L transform. Now K-L transform also refer to as the optimal transforms because essentially K-L transform obtains the transformation kernel based on the eigenvalues and eigenvectors of the covariance matrix. Therefore, in that sense it is optimal because it is based on the image itself that the transformation kernel is designed whereas the techniques like the discrete cosine transform, or rather to begin with it we have the Discrete Fourier Transform or what is called as the DFT DFT as applicable in the two dimensional space, then we have the Discrete Cosine Transform; this has become a very popular technique DCT in the domain of image compression because the DCT the advantage is that it is a real transform, its kernels are trigonometric kernels so it is based on the cosine values, the set of cosine values that are used to form the basis images using which you are going to represent any other image should be represented as a combination or a weighted combination of such basis images and the basis images are formed using the cosine transformation values so that is the DCT.

(Refer Slide Time: 56:19)



Transformation :

- K-L Transform
- Discrete Fourier Transform (DFT)
- Discrete Cosine Transform (DCT)
- Discrete Wavelet Transform (DWT)

Now, in very recent times, in the recent standard like the JPEG 2000 there the Wavelet the discrete Wavelet transform has also been used and this also is a very popular transformation process. So unlike the trigonometric transformations which use a waveform as the basis; waveform means using the cosine for example or using the exponential which is used in the......... I mean, the complex exponential which is used in the Fourier transform, now using waves................. that is based on the waves because it begins at minus infinity and retains at plus infinity, but in the case of Wavelets, Wavelets are basically the time limited signals which are used as the basis and using the Wavelets as basis the discrete Wavelets transformation techniques result, DWT and DWT also has been very popularly used in all the recent image compression applications.

This is about the basic background that I wanted to give you about the image coding and we will go into some detailed aspects of the Discrete Cosine Transform and Wavelet transform in the coming lecture, thank you.