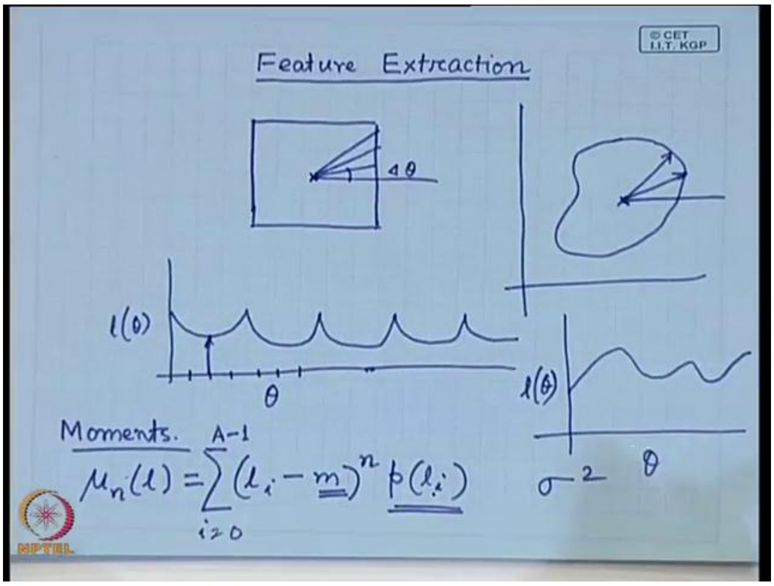
Pattern Recognition and Application
Prof. P.K. Biswas
Department of Electronics and Electrical Communication Engineering
Indian institute of Technology, Kharagpur

Lecture - 4
Feature Extraction – III

Good morning, so continuing with our lecture on feature extraction, in the last class we have talked about one of the boundary feature, which we have said as signature. And we have said that the signature is obtained something like this.

(Refer Slide Time: 00:38)



That a given a two dimensional shape, what you have to do is you have to find out the centroid of this two dimensional shape. And if I take the horizontal line as the reference then I have to find out the distance of the boundary points from this centroid by at different orientations. So, if I take the orientation at an interval of say delta theta something like this. So, this is delta theta, and then if I plot theta versus l theta where at every instant of an interval of delta theta I will get some distance value and if I have a square or a rectangular shape like this then this distance with respect to theta will vary in this form. So, it is like this.

In the same manner if I have some arbitrary shape again from the centroid of the shape you take the distance of the boundary points in different directions where this orientation this direction will be measured with respect to the line which is horizontal again I will get with respect to theta the distance of the boundary points from the centre which is given as l theta. And this l theta will also have some variation of this form.

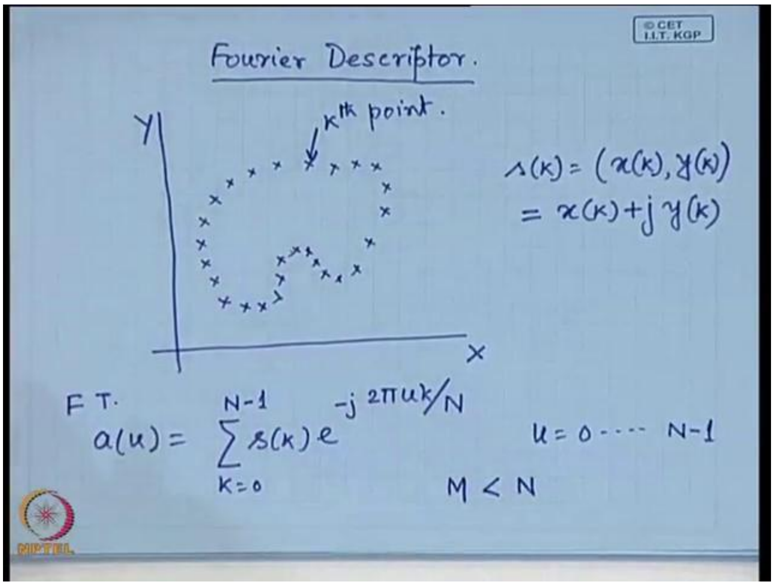
So, what I get is this 2 dimensional boundary information that is converted to a 1 dimensional function where it is a distance function distance with respect to orientation that is theta. So, if I take this as the boundary signature then I can generate different types of features or feature vectors from this particular pattern itself and 1 of them is the moments.

So, a moment of order say n that is given by mu n of l that will be nothing but l I minus m to the power n into p of l I take the summation for I varying from 0 to say capital A minus 1 where capital A is the number of samples that I have within this pattern. And what I have to compute is p of l i. So, how do I get this p of l i. If this distance values that we are getting that we quantize into a number of distance values. Usually this distance value will be a continuous value.'

So, what I have to do it is do is I have to quantize this distance values into a number of bins. And l i is the i th bin and once I quantize it then I have to find out that in i th bin how many distance values actually occur. So, for every i th bin I get the number of distance values occurring in that i th bin. So, effectively what I get is histogram of this quantized distance values. And if I normalize this histogram with respect to the number of samples that I have which in this case is a then what I get is some estimate of the probability of occurrence of the l i.

And then by using this expression I get the nth moment of the distance values are and m in this case is the mean of those distance values. And obviously you find that for n equal to 2 that is the second moment that has got a spatial significance because if I put n equal to 2 this simply becomes l i minus m square into p of l i take the summation which is nothing but the statistical variance which is sigma square. So, if I take the moments of different orders 2, 3, 4 and so on each of those moments becomes a feature which is a feature of this signature. So, these features can be used to represent this signature or it forms if I get a number of such features concatenated 1 after another I get a feature vector which represents this particular signature. So, this is also one of the boundary based features that can be used for recognition purpose.

(Refer Slide Time: 05:43)



Similarly, I can have other features the other 1 is Fourier descriptor. So, what is Fourier descriptor let us assume that we have a number of boundary points, so like this is in or 2 dimensions. So, I have this x dimension and y dimension. So, every k th point in this boundary say this is my k th boundary point is actually a coordinate. So, I can say that this is k th boundary point s k is actually having the coordinates x k and y k. So, for different values of k I get set of different boundary points I can assume that this coordinate can be represented by a complex number. So, I can represent this as x k plus j y k. So, it is simply my y coordinate I am assuming to be the y axis I am assuming to be the imaginary axis. So, this coordinate I represent by a complex number.

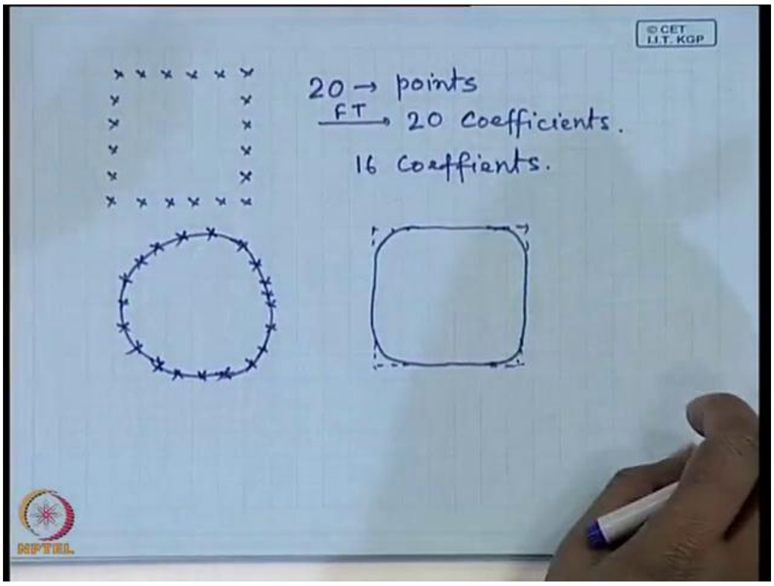
So, once I do this I can find the Fourier transform of this particular sequence of boundary points. So, this Fourier transform I will have. So, if I find out the Fourier transform I will get a u it is equal to s k e to the power minus j 2 pi u k upon capital N where k varies from 0 to capital N minus 1. And capital N is the total number of samples I have in this boundary representation. So, what I am simply doing is I have a set of 2 dimensional points which are nothing but the boundary points these points are represented as complex numbers. So, I have a set of complex numbers or a complex samples I take the Fourier transform of that set of complex numbers.

So, I get this a u where u will also vary from 0 to capital N minus 1 because if I take the Fourier transform of n number of points I get n number of coefficients. So, this u also varies from 0 to capital N minus 1 that means I get capital N number of coefficients. So, this capital N number of coefficients can be represented as a vector. So, I have an n dimensional vector however we know that when I take a Fourier transformation, the higher order coefficient gives us the detailed information of the pattern. And the lower order coefficient gives us average information of the pattern.

So, it is possible that if I am not much interested in the higher order coefficients or much detailed information within the pattern then I can truncate the vector. So, instead of considering all the n values of a u I may decide that I will take only m values of a u where m is much less than n, so that I can reduce the dimensionality of the feature vector. So, what is the effect if I reduce the dimensionality of the feature vector is that once I have taken the Fourier transformation and got the Fourier coefficients. If I take the inverse Fourier transformation then I should get back my original pattern or the original shape, but as the higher order coefficients I am removing from my feature vector.

So, whatever the number of features that I have in my feature vector, if I take the inverse Fourier transformation of that in that case the detailed information present within my original shape they will be lost, but the total number of points will remain the same because when I truncate. And I want to take the inverse Fourier transformation to take the inverse Fourier transformation all the coefficients which I have removed I set those coefficients values equal to 0. And take the inverse Fourier transformation, so that my total number of points remain the same. Then how does it affect when I in the shape.

(Refer Slide Time: 10:56)



So, for example, if I have boundary points something like this. So, it is a square boundary and there are 20 boundary points. So, square boundary having 20 boundary points. So, when I take the Fourier transformation Fourier transformation will give me 20 coefficients. So, out of this 20 if I decide that some of the higher order coefficient values I will remove.

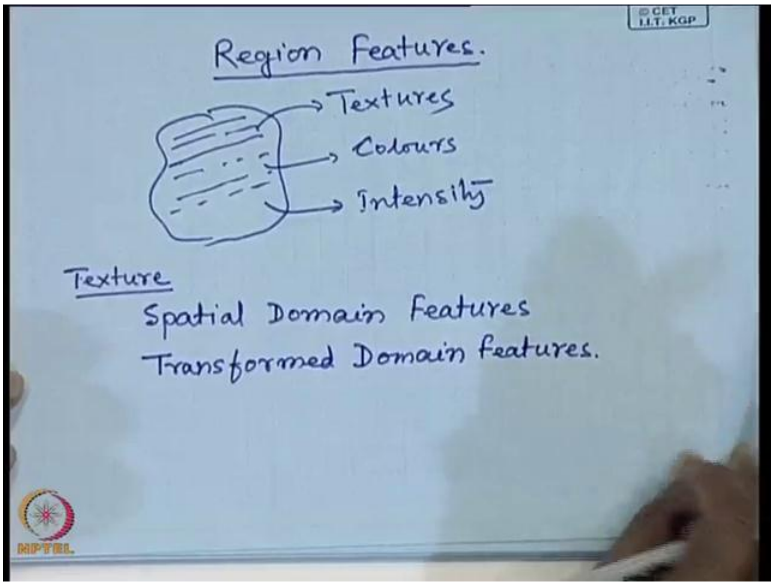
So, if for example, I decide that I will retain only 1 or 2 coefficients to represent to give me the Fourier descriptor, and using those 1 or 2 coefficients if I take the inverse Fourier transformation by setting the coefficients which are removed equal to 0. Then the kind of boundary after reconstruction what I will get is a circular boundary something like this. Though the number of points on the circular boundary will be equal to 20 I will have 20 points on this boundary.

So, all these 20 points will be there, but you find that all the detailed information which at there at the corners those detailed information are lost. The reason is very simple that if I take only 1 coefficient that simply gives you the d c value. So, your boundary will be an uniform boundary whereas, if instead of just 1 or 2 if I decide that I will retain say 15 or say 16n coefficients and I will take the inverse Fourier transformation by using those 16 coefficients the remaining 4 coefficients I want to make equal to 0.

So, reconstructed boundary in that case will be something like this though the total number of points on this boundary will remain the same as 20, but you find that these corner information which is not there anymore. So, over here these detailed information are lost, but overall shape is retained. So, in some cases we may feel that even this kind of description is sufficient for all recognition purpose. So, instead of taking all that 20 coefficients we may decide that I will take 16 coefficients or I will take 10 coefficients or even 8 coefficients as representation of this boundary.

And using those 10 coefficients I get feature vector of dimension 10. And this feature vector I will use for recognition purpose. So, these are the various boundary based description techniques. Now, let us go to the inside the boundary that what are the different kind of region descriptors or region based features that we can extract which can help us in the recognition process.

(Refer Slide Time: 14:44)



So, let us take some region features suppose I have a close boundary something like this. Now, far the descriptors that we have discussed those descriptors are based on the boundary information. That means it makes use of only the information which is given by the boundary I have not made use of any information which is their inside.

Now, there are different other type of features also which are shape features or boundary based features and there are n number of such different features. So, I am not going to discuss all of them, but just the concept that I can obtain features only from the boundaries and this simply gives you the shape information it does not tell you anything what is the inside the shape.

Now, coming to the region features this region can have different kinds of textures. So, this bounded region can have different types of textures or it can have different types of colours or different regions if it is a black and white image un textured that is a uniform region can have different intensity values. So, this intensity value itself can be 1 feature the colour information of the region can be 1 feature as well as texture of that region can be another feature.

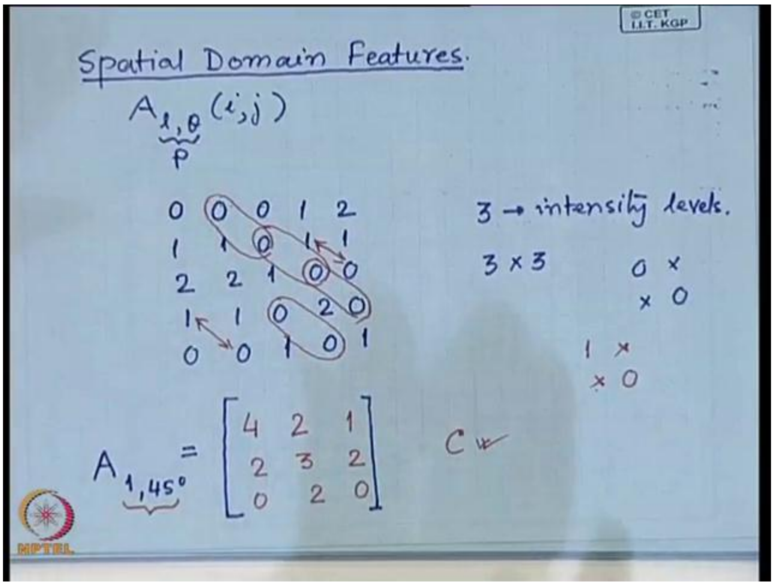
So, I can obtain the descriptors or the feature vectors corresponding to either feature corresponding to colour or even corresponding to the intensity value. So, if it is intensity value it is a scalar function. So, I get a single feature it is not a feature vector, but that feature can be combined with other features to give a feature vector.

So, let us first consider what are the different types of texture features, that I can obtain from a textured region. So, I will talk about the other extraction techniques like colours later on, but first let us talk about the texture feature. So, when I go for feature region feature extraction I can have 2 types of approaches 1 is I can extract the features using the spatial domain information itself or even I can do some sort of transformation. And I can take the transform this feature

So, like this what I have shown in this case with respect to signatures over here the signature is a pattern. So, if I take the samples on this pattern that itself becomes a feature vector or I can take the different moments of this that becomes a feature vector the moments become a feature vector. If I take the Fourier transformation of this pattern the Fourier transformation coefficients become a feature vector.

So, either I can take the features from the spatial domain that is from the raw data or I can also extract the features after taking transformation of the raw data. So, I can extract both spatial domain features as well as transformation domain features. So, I will have either spatial domain features or I can have transformed domain features. So, let us first see that what kind of spatial domain features that can be obtained from a texture.

(Refer Slide Time: 19:01)



So, when we talk about spatial domain features particularly in case of texture the features are obtained from a matrix which is called co occurrence matrix. So, co occurrence matrix is defined like this say A l theta i j where l theta taken together defines a parameter which is called a position parameter p, l tells you what is the distance and theta tells you what is the direction. And i and j are 2 intensity values as we have seen it is a co occurrence matrix it tells you that what is the frequency at which an intensity value i and an intensity value j will occur together, where the position of i is defined by this position parameter p with respect to intensity value j is that.

That means I will have in the image intensity value i or i th intensity value which is at a distance l in the direction theta from another pixel having an intensity value j. So, I have to count that how many times this i j pair following this position parameter p occur within the image. So, that count that number gives me an element a i j in the matrix A l theta p. So, after obtaining this matrix the number of counts in which this intensity pair i and j following this position parameter p occur within a given image.

And if I normalize that then what I get is the frequency of occurrence or the probability of occurrence of the intensity pair i j within the image following this position parameter p. So, let us take very simple example suppose I have a sample image something like this. So, suppose this is my sample image and I want to determine this particular matrix

A and suppose the value of a l is 1 that means I have to find out the pixel pairs intensity value pairs at distance 1 and at an angle of 45 degree. So, how will be this matrix.

Now, here you find that there are 3 different intensity levels 0 1 and 2. So, these are the 3 different intensity levels. So, naturally this matrix A will be of dimension 3 by 3 because it is indexed by intensity values i and j, i and j are nothing but the intensity values. So, the matrix A will also be of dimension 3 by 3. Now, to compute the 0eth element A l theta 0 0 both i and j they are 0. So, I have to find out the 0 0 pair occurring like this at a distance of 1 and inclined by an angle of 45 degree. So, when you come to this particular image sample image you find that this kind of combination I have over, one over here, one over here right, one over here is there anymore and of course.

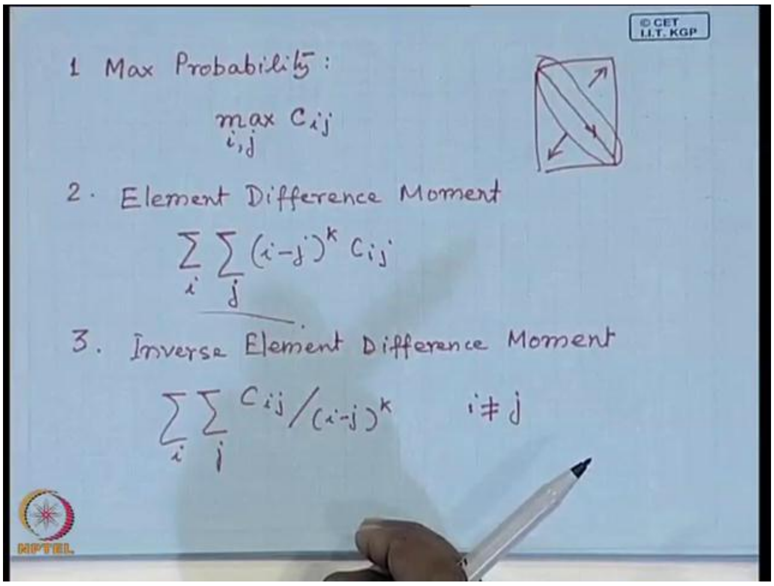
So, I have 4 such occurrences where this 0 0 intensity value they appear following our position parameter 145 degree. So, A 0 0 will have a value 4 over here then coming to 0 1 I have to have a pattern of this form 1 0 I do not bother about these 2 locations because my position parameter is 1 direction is 45 degree and 0 1 means the occurrence of 0 with respect to 1 following this position parameter. So, the kind of intensity pairs that I have to have is 1 0 like this I do not bother about these 2 whatever be the value of these 2 that is immaterial to me.

So, you find that how many such occurrences I have within this image 1 0, 1 is over here 1 0 and 1 is over here. So, this value will be 2. So, if you compute like this all the pairs where the intensity values are 1 of 0, 1. And then you get this matrix as four 2 1 2 3 2 0 2 0. So, this is matrix A A l theta i j where this is the kind of occurrences of the intensity pair that I will have.

Now, from this matrix I compute the co occurrence matrix C which is nothing but normalization of this matrix l theta with respect to total number of occurrences. So, if I take the sum of all these elements in the matrix theta that gives you the total number of occurrences of the intensity pairs. So, divide this A by total by the sum of the elements in this matrix A I get this co occurrence matrix C.

So, this co occurrence matrix C simply tells me, what is the frequency of occurrence of different intensity pairs following this parameter position parameter p. So, I can have multiple numbers of such co occurrences co occurrence matrixes for multiple position parameters because I can define any position parameter. And for every such position parameter I will get a co occurrence matrix. So, find what do you get the information that you will get from this occurrence matrix because you are looking for the intensity pairs which occur within the image following some position parameter. So, it gives you various information like what is the regularity what is the interval at which the different intensity pairs they occur. So, various such information can be obtained from the occurrence matrix. And the kinds of features which are actually obtained which are computed from the co occurrence matrix are given by this.

(Refer Slide Time: 28:01)



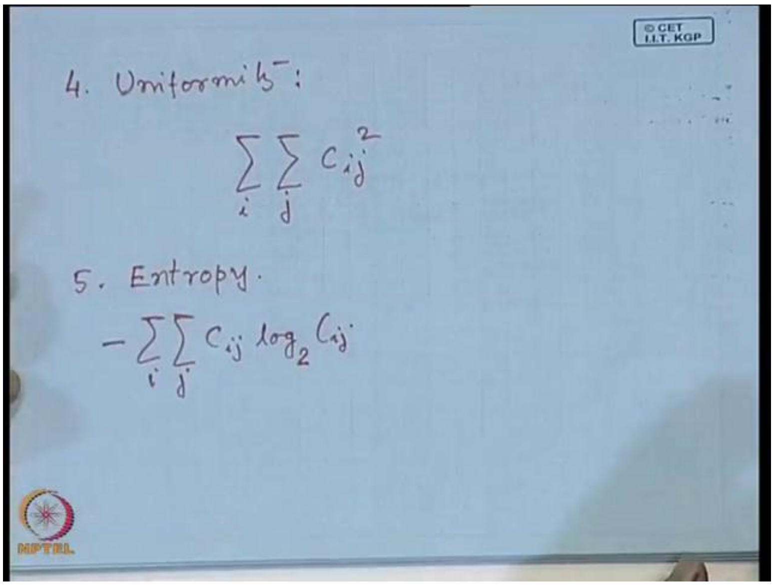
One is the maximum probability. This maximum probability is nothing but max of C i j where the maximum is taken over i and j. So, this maximum value tells you that what the pattern predominant within the texture is. So, if I take different samples of the same texture then this max value will remain the same more or less same. So, this gives me an indication that how I can identify 2 patterns or 2 samples of the same pattern or 2 samples of the same texture. If the texture are different textures are different then this max C i j will also be different I will get a maximum, but for different values of i and j not for the same values of i and j. So, this is 1 of the features that can be obtained from the co occurrence matrix.

The second type of feature that can be obtained from the co occurrence matrix is element difference moment which is defined as i minus j to the power k C i j take the summation of this over i and j. So, what does this feature tell you find that you are taking i minus j to the power k into C i j. So, naturally these value will be minimum whenever i and j are same that is whenever i is equal to i is equal to j means I have this matrix i is equal to j means the elements on the main diagonal of the matrix.

So, when I compute this the value will be very low if most of the C i j or the maximums of the C i j they appear along the main diagonal. If most of C i j are away from the main diagonal for all the elements away from the main diagonal the values of i minus j is high. So, the value will be more the summation sum of value will be more whereas, if C i j non most of the non 0 C i j or maximum values of C i j they appear along the main diagonal then this value will be more quite less. So, this is another feature describing the textures.

The other can be just inverse of this which is called inverse element difference moment, where it is defined as C i j upon i minus j to the power k take the summation over i and j. And naturally this is not defined for i equal to j, because i equal to j i minus j to the power k becomes 0, so this term becomes undefined. So, it will have this inverse element difference moment will have just an inverse effect of element difference moment wherever element difference moment is high inverse element difference will be low and wherever the other 1 is low this 1 will be high. So, this gives you an indication of how the co occurrence matrix of a given texture look like. And if I have samples from the same co occurrence matrix from the samples of the same texture the co occurrence matrices will also be similar similarly, the other feature.

(Refer Slide Time: 32:44)



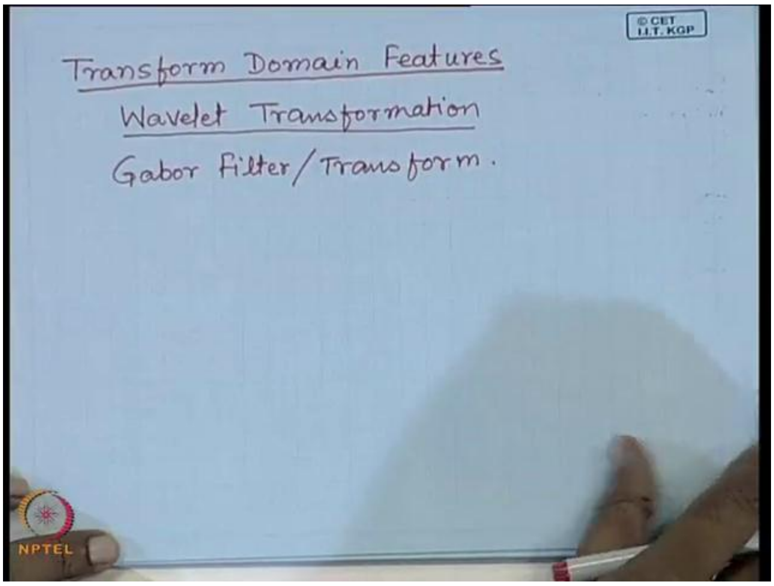
That can be obtained from the co occurrence matrix is what is called uniformity. And uniformity is simply defined as summation of C i j square you take the summation over i and j. And you will find that this value will be higher if all the C i j are similar or same. That means all the intensity pairs they are equally likely within the given image. So, that becomes a regular pattern. And the other features which is typically obtained from this C i j because C i j gives you the joint probability of occurrence jointly i and j, how do they occur following the position parameter. And whenever I have a probability measure I can define something called entropy.

So, I can also find out what is the entropy and as you know that the definition of entropy is simply given by C i j then log with respect to base 2 of C i j take the summation over I and j and negate it. So, if every C i j value is random the entropy will be very high that means this value will be quite high if the C i j values are random whereas, if the C i j values are more or less same the entropy value will be very low.

So, the entropy of a source which generates random symbols is quite high the entropy of a source which generates the same symbols or the symbols which can be predicted from the previous symbols that is very low. So, these are different kinds of features we can obtain in the spatial domain by making use of the co occurrence matrix of the texture. There are various other ways to generate the spatial domain features also, but I am not going into details all of them in fact there are numerous ways in which, the feature vectors can be generated.

And the way you generate the feature vector that depends on what kind of application, you have maybe that for a kind of application you will be, you may find out, that you can generate some sort of feature which is not really even in textbooks that is also possible. So, what kind of feature you use for a particular application that depends on the application or that depends on the kind of objects that you have. Now, let us see that, what are the different kinds of transform domain features that can be obtained for textures.

(Refer Slide Time: 35:44)

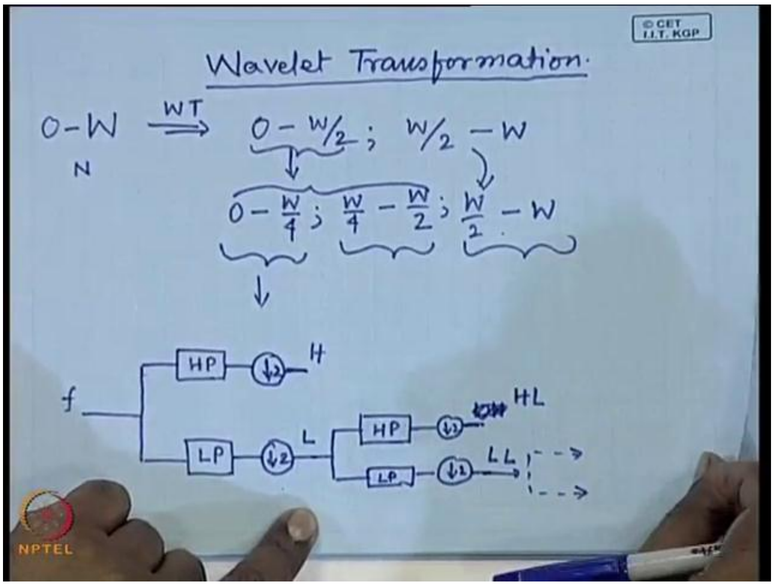


So, let me go for the transform domain features. So, whenever we talk about transformation the first kind of transformation that comes into mind is the Fourier transformation, or that cosine transformation the difference between Fourier transformation and the cosine transformation is that Fourier transformation gives you the complex coefficients. So, you also have phase information embedded into the Fourier coefficients the cosine transformation or the discrete cosine transformation gives you the real coefficients. So, you lose the phase information, however both Fourier transformation and cosine transformation they are very popular in signal processing.

Now, the problem with textures is that textures are some sort of random pattern they have both high frequency comp1nts as well as low frequency comp1nts. So, if I take the Fourier transformation or the discrete coefficient or the discrete cosine transformation of the textures. The Fourier coefficients or the DCT coefficients they do not give you much of discriminating power because I have all high frequency components as well as low frequency components appearing in different degrees in different types of transformations.

So, the kind of transformation which has become very, very popular for characterizing or for describing textures is, one is wavelet transformation. And the most popular to represent textures or to describe textures is gabor filter or gabor transformation. So, let us first talk about the wavelet transformation. For those who have done my image processing course wavelet transformation was not covered I do not know whether others have done it have you done wavelet transformation.

(Refer Slide Time: 38:15)



So, let us see what is wavelet transformation when you take the Fourier transformation or cosine transformation what is it that you do effectively anybody have the answer what does this coefficients mean whether it is Fourier coefficients DCT coefficients ((Refer Time: 39:50)) that is what you effectively get, but what does the transformation give you.

Whatever is the set of data you have you represent that set of data by a set of sinusoidal waves. So, if you have one dimensional signal you have one dimensional sinusoid, if you have two dimensional signal like images you have two dimensional sinusoid. For Fourier transformation I get two coefficients one is the cosine coefficient other one is the cosine coefficient the cosine coefficient gives you the real part and the sine coefficient gives you the imaginary part.

These two taken together gives you the phase information when you take the cosine transformation I get only cosine coefficients I do not get the sine coefficients. So, effectively I lose the phase information. So, when I take Fourier transformation or cosine transformation as I represent the data by a set of sinusoidal waves these sinusoidal waves are infinitely extended. So, what I have is I can equivalently say that I have wave transformations even a set of samples I am representing them by a set of sinusoidal waves of different frequencies and different magnitudes different amplitudes.

In case of wavelet transformation it is not wave it is wavelet that means part of wave or a small wave. So, in case of wavelet transformation the same signal is represented by tiny wave portions. Now, how I do not want to go into details of that because wavelet transformation will itself will be full time semester course, but I will simply touch up on how you implement wavelet transformation and what you get out of wavelet transformation.

So, when we talk about the wavelet transformation it is something like this suppose I have a signal let us assume I have one dimensional signal initially. And suppose the signal has a bandwidth of say w, what wavelet transformation does is wavelet transformation breaks this signal into sub bands the total bandwidth is w. So, let us assume that it is 0 to w when I apply wavelet transformation wavelet transformation breaks this bandwidth into two sub bands one is 0 to w by 2 and other sub band is w by 2 to w.

When I continue further this 0 to w by 2 is further converted into 0 to w by 4 half of this. And the other sub band will be w by 4 to w by 2 and this one is retained w by 2 to w. So, I am always dividing the signal into a number of sub bands. And the same operation is repeated in the lower sub band. So, effectively what I get is a tree kind of structure.

Now over here you find that I have the original signal of bandwidth 0 to w when I convert this into a signal of bandwidth of 0 to w by 2 naturally the detailed information present in the signal is lost because high frequency components I am removing. So, this 0 to w by 2 it is a coarser version or a signal same signal at a lower resolution. Whereas, the higher sub band w by 2 to w that gives you the detail information which is present in the...

So, I can represent this kind of filtering in the form of form like this. So, I have 2 filters one is a high pass filter of cut off frequency w by 2 and I have a low pass filter of the same cut off frequency w by 2. And this is my input signal say f I retain the output of the high pass filter. And this output of the low pass filter that again I want to sub divide into 2 sub bands as has been d1 here 0 to w by 2 has been divided into this sub band whereas, the higher sub band w by 2 to w that has been retained.

Now, there is information here that my original bandwidth was 0 to w. And from ((Refer Time: 44:14))sample rate you know, that when you have a bandwidth of w the minimum sampling frequency has to be twice of w that is what is your sampling ((Refer Time: 44:25)) sampling rate otherwise you lose the information.

So, if I have total n number of samples within this to represent this particular signal. Now, because I am reducing the bandwidth to w by 2, so following the ((Refer Time: 44:47)) t sampling rate my sampling frequency comes down by a factor of 2. So, I have to retain total n number of samples, but by using n by 2 numbers of samples I retain the same information. So, similarly here... So, there is a concept of sub sampling by a factor of 2.

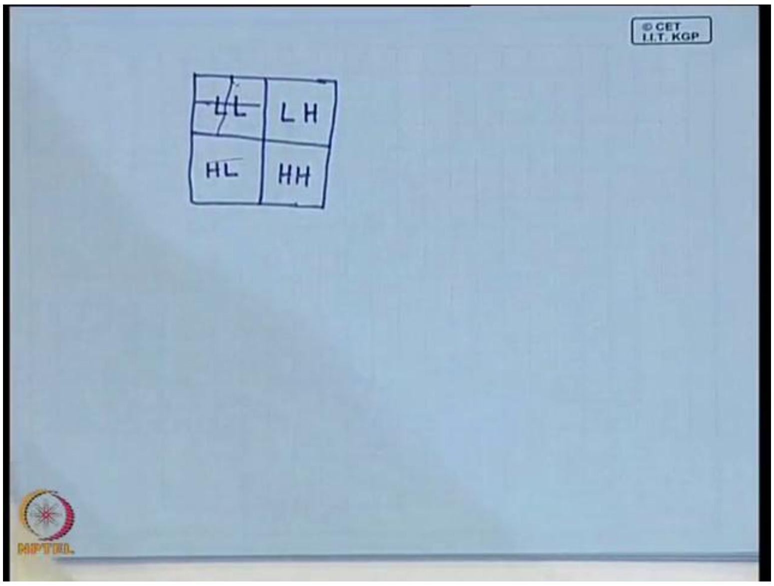
So, we find that when I have total n number of samples if I simply do the filtering low pass filtering and high pass filtering the output of this band will be n number of samples this band will also be n number of samples. So, total from n samples I generate 2 n samples, but I can reduce the size by using this concept of sub sampling by a factor of 2. So, this also reduces to n by 2 numbers of samples this also reduces to n by 2 numbers of samples.

So, the total number of samples that I retain is again n. So, I go for this n by 2 sub sampling and this 1 I called as high frequency band and this is the low frequency band. Now, I apply the same filtering operation over here high pass filter over here low pass filter over here because I am going to further subdivide the signal into different sub bands. So, this is low pass filter this will again be sub sampled by a factor of 2, this will again be sub sampled by a factor of 2. So, that way your total numbers of samples remain the same.

And this sub band now I call as LH because this was let me reverse this notation I call it HL because this sample was this sub band has been obtained by first low pass filtering of the original signal. And that sub band further high pass filtered. So, first low pass filtered then high pass filtered. And this sub band I call as LL this is low pass filtered low pass filtered.

So, I can continue with this further to give me different levels of decomposition. And this set of coefficients that I get in different sub bands this is what my wavelet transformation coefficients are. So, this is what I have in case of one dimensional signal what do I have in case of images because image in a 2 dimensional signal. So, when I have images and that has two dimensional that is a two dimensional signal I can do high pass filtering horizontal direction I can do low pass filtering in horizontal direction, I can perform high pass filtering in vertical direction, I can also perform high pass filtering in vertical direction. So, I actually have 4 different combinations low pass filtering horizontally low pass filtering vertically high pass filtering horizontally low pass filtering vertically low pass filtering horizontally high pass filtering vertically high pass filtering horizontally high pass filtering vertically.

(Refer Slide Time: 48:01)

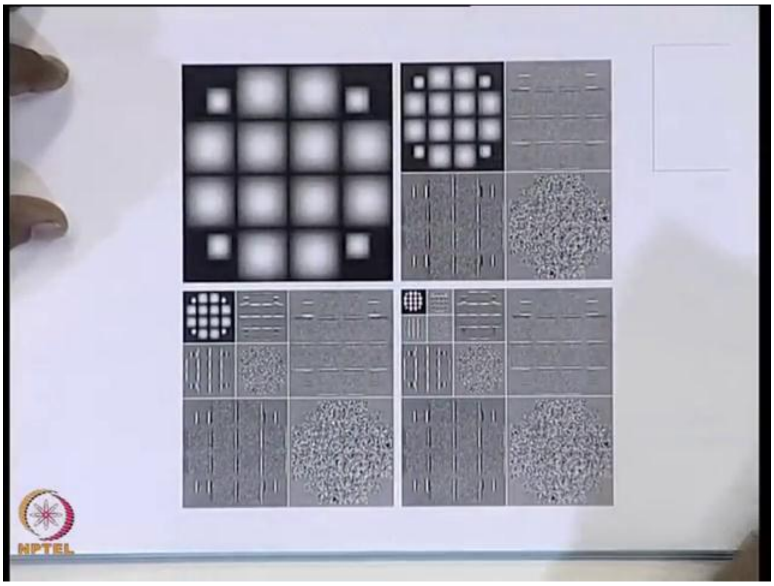


And all this different sub bands if I put in this form because every time I decompose by using sub sampling I reduce the image size by 4 half in the horizontal direction half in the vertical direction. So, you over here I put the sub band which is low pass filtered horizontally low pass filtered vertically. So, this is nothing but our reduced resolution version of the same image. Over here I can put that particular sub band which is high pass filtered horizontally low pass filtered vertically. Here I can put the sub band which is low pass filtered horizontally high pass filtered vertically and here I put HH band.

Now, coming to different sub bands this is nothing but lower resolution of the same image what is this I am doing high pass filter horizontally low pass filter vertically high pass filtered horizontally means it will try to enhance all the vertical edges you are taking differentiation in the horizontal direction. So, all the vertical discontinuities will be highlighted. So, mainly the high vertical edges will be highlighted in this LH sub band.

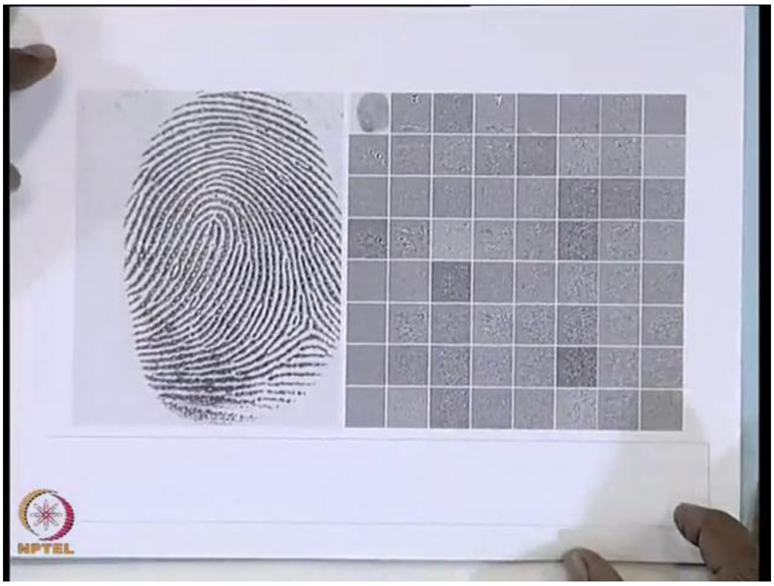
Similarly, HL sub band will highlight all the horizontal edges right HH which is high pass horizontal high pass vertical that will mostly highlight all the diagonal edges right then I perform further subdivision on this LL sub band go continue with this every time I get sub bands at different resolutions. So, that is what wavelet transformation? There is another concept which is called wavelet packet. Wavelet packet is I go for decomposition of each of these sub bands in case of wavelet only the LL sub band you go on decomposing in case of wavelet transformation packet. Wavelet packet transformation, you sub divide each of the sub bands. So, I get a set of different sub bands let me show an example of this.

(Refer Slide Time: 50:33)



So, this is an image suppose this is my original image after first level of decomposition I get this lower resolution version of the same image. Over here I have all the horizontal edges here the combination was different high pass filtered horizontal low pass filtered vertical it does not matter whether I put that sub band here or I put that sub band here. It is simply position and this is high, high HH sub band. So, this is what I have after first level of decomposition. After second level of decomposition this is what I get, because then this LL sub band is further sub divided after third level of decomposition this is what I get this LL sub band is further sub divided. Similarly, in case of wavelet packet transformation what I get is something like this.

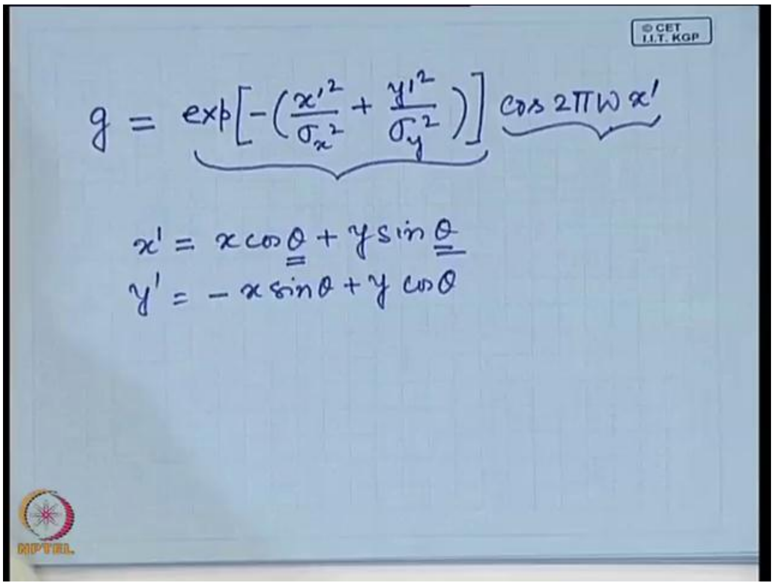
(Refer Slide Time: 51:32)



This is a finger print image. So, you find that every sub band has been sub divided further. So, the size of every sub band is same after wavelet packet decomposition whereas, after wavelet decomposition as I go higher up in this tree the sub band size becomes lower and lower every time it is 1 fourth of the sub band size of the previous level. So, effectively what I do is given a signal I divide the signal I have broken the signal into a number of sub bands.

Now, depending upon the type of the signal the energy of the coefficients in the energy sub bands will be different all the signals will not have the same energy in the same frequencies. So, if I compute the energy of these different sub bands put them in a particular order. Then this ordered arrangement of the energies of different sub bands that itself gives me a feature vector right, which can be used for pattern recognition purpose. Similarly, the other transformation that I was talking about is gabor filtering. A gabor filter is given by an expression like this.

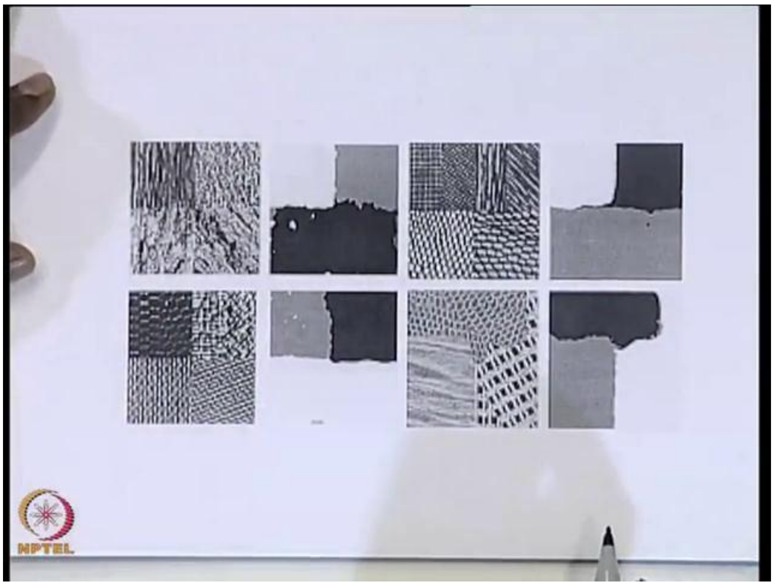
(Refer Slide Time: 53:05)



Exponentiation of minus x hat square by sigma x square plus y times square by sigma y square into cosine 2 pi omega x time. So, this is my expression of the gabor filter. So, here you find that this term is nothing but a Gaussian envelope in 2 dimensions centred at 0. And this is a cosine term which actually modulates this Gaussian envelope. And this x prime and y prime are given by x prime is equal to x cosine theta plus y sine theta and y prime is minus x sin theta plus y cosine theta where theta is the orientation of this Gaussian envelope.

So, if I convolve my image if I do filtering of my image using this Gaussian filter for different orientations for different values of omega and different values of this spread sigma x and sigma y I get different sets of coefficients. And that different set of coefficients can give me the feature information. And this particular filter is very, very popular for texture classification as well as texture segmentation.

(Refer Slide Time: 55:14)



Here, I want to show you just some result texture classification result that has been obtained using this sort of gabor filter coefficients. So, here you find that this is a combination of three textures 1 texture over here, one texture here and though here it appears to be two different textures, but it is the same texture, one is the rotated version of the other. And after classification using this gabor feature you find that this part has been put into one class this part has been put into another class, and this entire part has been put into another class similarly, in other examples. So, this clearly shows that gabor filter coefficients have high discriminating power of the textures and. So, this gabor filter coefficients can be used for texture descriptor and as well as for texture recognition purpose. So, we will stop here today.

Thank you.