Computer Vision and Image Processing - Fundamentals and Applications

Professor Dr. M K Bhuyan
Department of Electronics and Electrical engineering
Indian Institute of Technology, Guwahati

Lecture-25

Image Texture Analysis - II

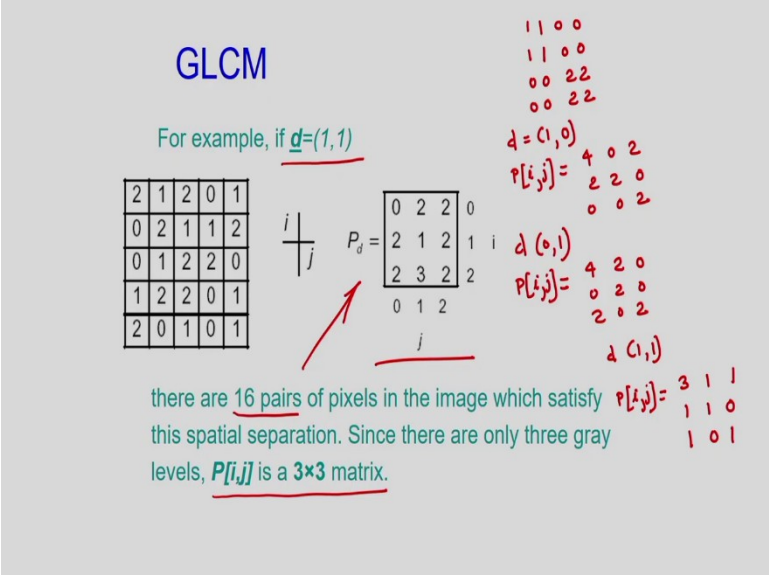
Welcome to NPTEL MOOCs course on Computer Vision And Image Processing - Fundamentals And Applications. In my last class I discussed the concept of image texture analysis. So, what is texture? Texture means spatial distribution of gray level intensities, I discussed 4 research directions one is texture classification, one is texture synthesis, one is texture segmentation and one is Shape from texture. A particular texture can be represented or described by 3 techniques I highlighted, one is the structural method, one is the statistical method and one is the spectral method.

In case of the structural method, I have to find the placement rule for the texels, texel means the basic unit of a texture, that is the primitive. Texel means a group of pixels having homogeneous property and based on this texel I have to find the placement rule for the structural representation of a particular texture. The next one is the statistical method. So, I have to extract some statistical parameters and based on these parameters, I can represent a particular texture.

And finally, I discussed the technique is the spectral method. In spectral method, I have to determine the Fourier transform of the image. And based on the Fourier transform of the image, I can represent a particular texture. Last class I discuss the statistical methods for texture representation. So, for this I can extract some statistical parameters like mean, standard deviation, third order moment like this I can extract and based on this I can represent a particular texture.

After this I discussed the concept of GLCM or gray level co-occurrence matrix. So, today I am going to discuss some other texture descriptors. So, one is the autocorrelation function that I am going to discuss today. Another one is very important, that is the Gabor filter based texture representation. So, I can extract the texture features by Gabor filter and finally, I will discuss the concept of local binary pattern, LBP. So, how to represent a particular texture by using LBP. So, let us see the concept of the GLCM that I discussed in my last class.

(Refer Slide Time: 3:08)



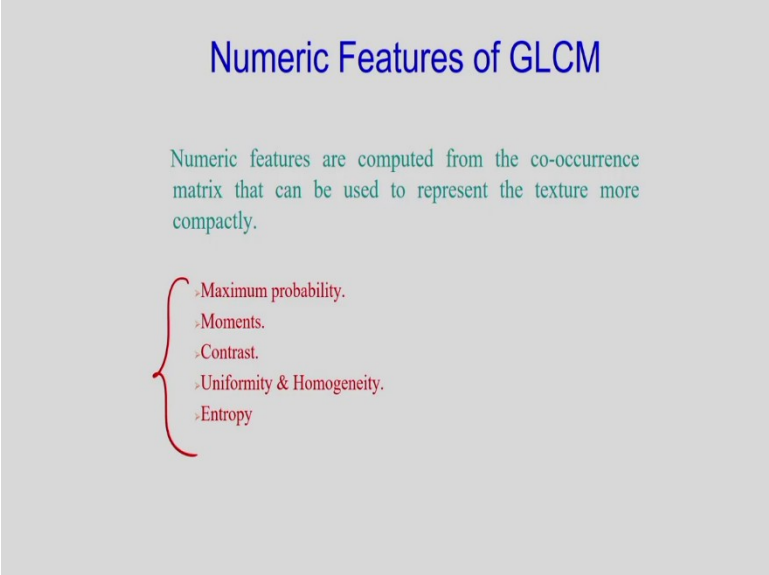
So, you can see in my last class I discussed this, the concept of the GLCM. So, from the input image, I can determine the GLCM, that is that gray level co-occurrence matrix. And in this case I am considering 3 gray levels. So, that is why the P_{ij} will be 3 by 3 matrix and for GLCM I have to define the displacement vector. So, the displacement is 1 and 1. Suppose I can consider another example of GLCM suppose, my input is 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 2, 2, 0, 0, 2, 2.

So, this is my input image and, in this case, I will be getting the 3 by 3 gray level co-occurrence matrix. So, suppose my displacement is 1, 0 in x direction it is 1 and in y direction it is 0. So, corresponding to this d (1, 0) my array the array is P_{ij} the P_{ij} will be there you can verify this 4 0 2, 2 2 0, 0 0 2. So, this is my array also if I consider the displacement suppose, and d (0,1) corresponding to the displacement my P_{ij} will be that is the gray level co-occurrence matrix 4 2 0, 0 2 0, 2 0 2, that is for a displacement d (0,1).

And suppose if I consider a displacement d (1, 1), that is in the x direction it is 1 and in y direction it is 1. So, corresponding to this displacement, the P_{ij} will be 3 1 1, 1 1 0, 1 0 1. So, you can verify this. So, what is the method, so, for all the displacements, so, different displacements, I have to find the GLCM. That is GLCM is computed for several values of d and the one who is maximize a statistical measure computed from P_{ij} is finally used. So, that is the concept of GLCM and also in my last class I discussed about the normalization procedure.

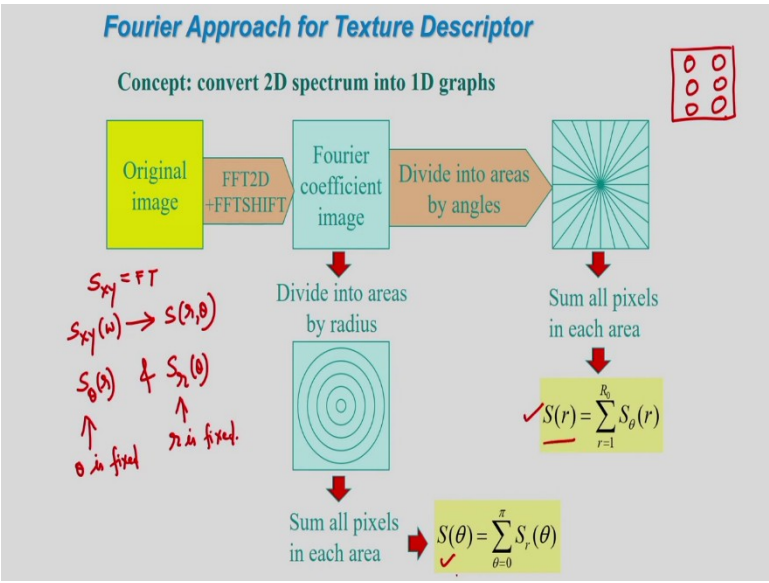
So, we have to do the normalization of the gray level co-occurrence matrix. So, in this case, in this example, if you consider this example, what I have to consider, these elements, we have to divided by 16 because 16 pairs of pixels in the image satisfy the spatial separation condition. So, that is why 0 is divided by 16, 2 is divided by 16 like this I have to consider corresponding to this co-occurrence matrix. So, I have to do the normalization.

(Refer Slide Time: 6:24)



And from the GLCM I can extract some features, the statistical features, the parameters are like this. The maximum probability I can determine, the moments I can determine, contrast I can determine, uniformity and the homogeneity also I can determine from the GLCM and finally, one important parameter that is the entropy also I can determine.

(Refer Slide Time: 6:50)



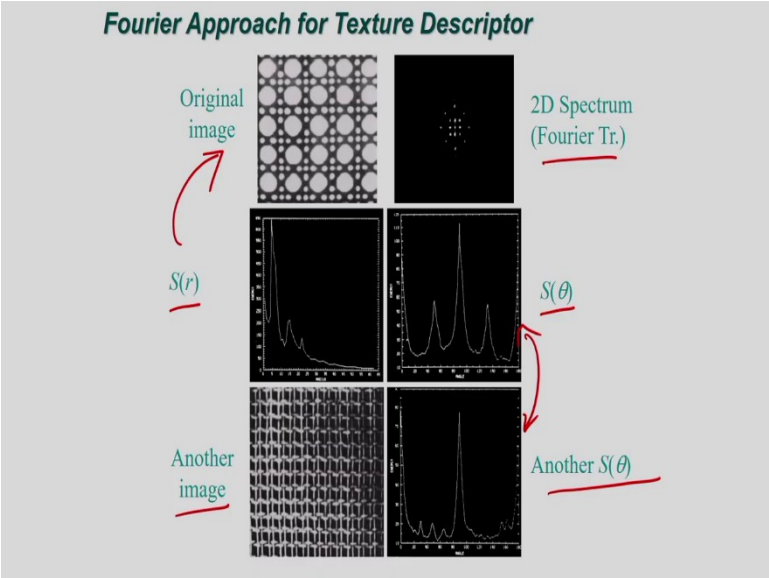
After this I discussed in my last class the concept of the spectral representation of a texture. So, for this, you can see the texture is nothing but repetitions. So, if I consider a texture, it is nothing but the repetitions of this texels. So, that is regularity in the texture. So, for this what I can do, I can determine the Fourier transform of an image. So, suppose S_{xy} is the Fourier transform of the image. So, in the Fourier transform, the prominent peaks in the Fourier spectrum give the principal directions of that texture pattern.

So, that is the first point, that is the prominent peaks in the Fourier spectrum give the principal directions of that texture pattern and also the location of the peaks in the frequency plane gives the fundamental spatial period of the patterns. So, this is the concept of the Fourier transformation. This S_{xy}, there is a Fourier transformation, the 2d Fourier transform of the image that I can represent in the polar form S (r, theta). I can convert to the polar coordinate.

Last class I discussed about this and I can consider this case S (r, theta) and another one is S_r theta I can consider. The first case is what is the first case, the theta is fixed and, in this case, I want to find a behavior of the spectrum along the radial direction. So, this is the theta is fixed and I want to see the behavior of the spectrum along the radial directions. In the second case, the r is fixed and I can see the behavior along the circle centered on the origin.

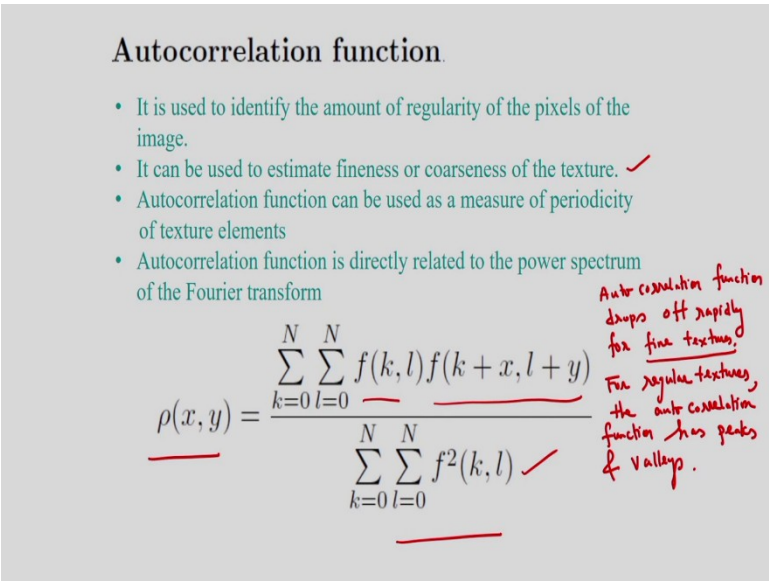
So, because the theta is variable here, r is fixed, that means, I can see the behavior along a circle centered on the origin that I can see. And based on this I can compute S_r and the S (r, theta), can compute. So, some are some all pixels in each area and some all pixel in each area I have to consider that compute S_r, I am determining, another one is S (r, theta) I am determining.

(Refer Slide Time: 9:15)



And you can see this example corresponding to this original image I am determining the 2D Fourier transform, there is a 2D Fourier transform I am determining and you can see these 2 spectrums, one is S_r and other one is S (theta) corresponding to this image. And if I consider another image the textured image corresponding to this image, you can see the spectrum S (theta) So, S (theta) is different and if you see these 2 spectrums, it is quite different. So, that means based on S_r and S (theta), I can represent different types of textures. This is about the Fourier approach for texture representations.

(Refer Slide Time: 10:02)



Now, next one is the autocorrelation function I can consider. So, in this case, one important property of textures is the repetitive nature. Repetition is due to the placement of the texture elements, the texture element is the texels in the image. So, that is the repetition due to the

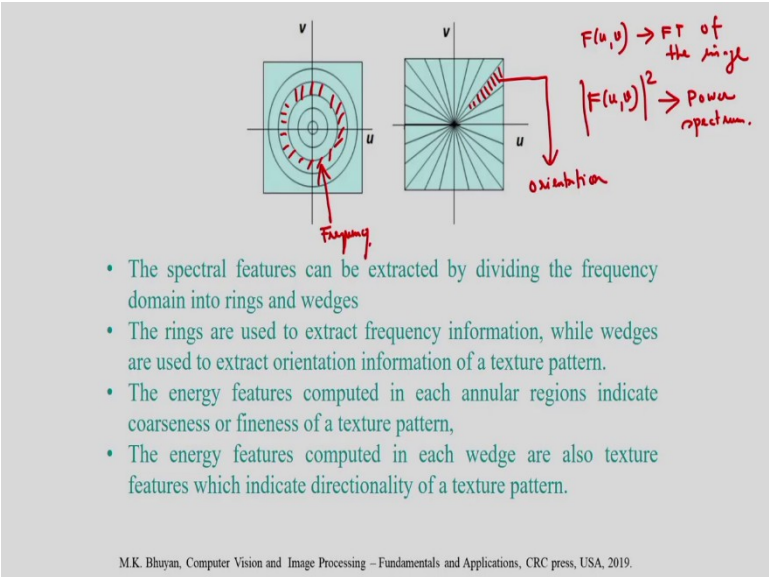
placement of texture elements, Texels in the image. So, in this case, this autocorrelation function, it is used to identify the amount of regularity of the pixels of the image and it can be used to estimate the fineness and the coarseness of the texture, that is one important point.

This autocorrelation function can be used to estimate fineness and the coarseness of the texture. An autocorrelation function can be used as a measure of periodicity of texture elements and autocorrelation function is directly related to the power spectrum of the Fourier transform. This autocorrelation function is defined like this corresponding to the image the image is $\rho(x, y)$. So, this is the definition of the autocorrelation function. So, f (k, l) I am considering f (k + x, l + y) and after this we have to do the normalization this is the $f^2(k,l)$.

So, this is the definition of the autocorrelation function. This autocorrelation functions drops off rapidly for fine textures. So, I can write like this autocorrelation function drops off rapidly for fine textures and for regular textures the autocorrelation function the autocorrelation function has peaks, autocorrelation function has peaks and valleys. So that means the autocorrelation function can be used as a measure of periodicity of texture elements.

And here I have shown the 2 cases, the autocorrelation function drops off rapidly for fine texture. So, for fine texture it drops of rapidly and for regular textures, the autocorrelation function has peaks and valleys. And also, the autocorrelation function can be used as a measure of periodicity of texture elements.

(Refer Slide Time: 13:04)

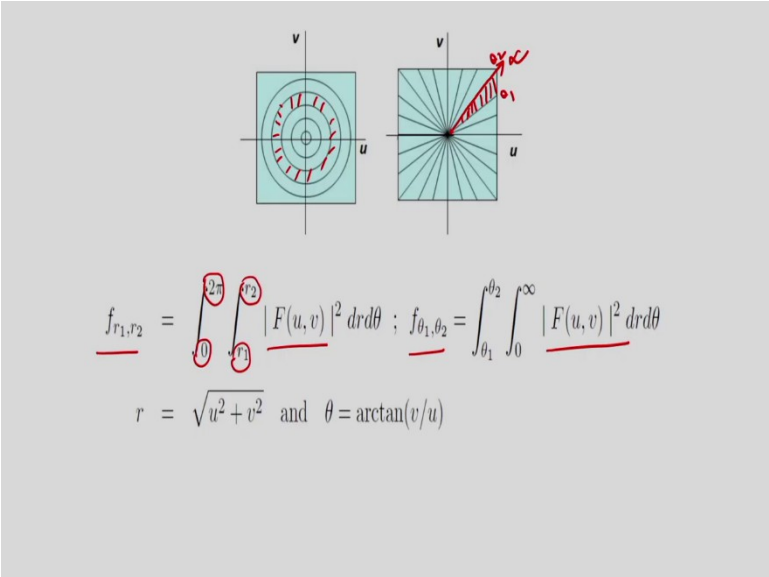


Now, suppose if I consider f (u, v) is the Fourier transform, the 2D Fourier transform of the image and ¿$f(u,v)$¿^{2}¿ is it is called the power spectrum. So, the spectral features can be extracted by dividing the frequency domain into rings and wedges. So, in the picture you can see, I am considering some rings, you can see the rings, these are the rings. I am considering the Fourier transform, the spectral features I can extract by dividing the frequency domain into rings or maybe the wedges suppose if I consider, this is the wedge, I can consider.

The rings are used to extract frequency information, while the wedges are used to extract orientation information of the texture pattern. So, the rings these are rings are used to extract the frequency information. And this the wedges I am using to extract the orientation information. The wedges are used to extract the orientation information. And after this, the energy features can be computed in his annular regions I can compute and it indicates the coarseness or fineness of a texture pattern.

So, I can determine the energy, I can determine the energy for the annular regions and it indicates the coarseness or the fineness of a texture pattern. Also, I can determine the energy features in each wedges and these features indicates the directionality of a texture pattern. So, that means, I can determine the energy features in the annular regions and also, I can determine the energy features in wedges.

(Refer Slide Time: 15:34)



And in this wedge, you can see, I am considering the power spectrum, so this is the power spectrum. In the first case I am calculating $f_{r_1 r_2}$ I am calculating. So, in this case what I am considering, that means, all the angles I am considering that is for the annular rings, so, this

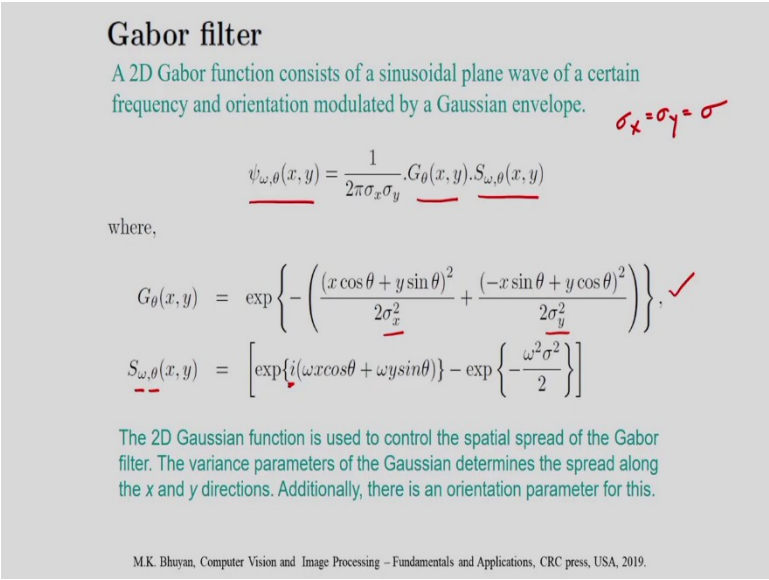
ring I am considering these the annular rings So, the angle is from 0 to twice pi I am considering, so this angle is from 0 to twice pi, that I am considering and but radius is from r 1 to r 2 I am considering.

The annular rings I am considering and I am determining the feature, the feature is $f_{r_1 r_2}$. In the second case what I am considering $f_{\theta_1 \theta_2}$ and, in this case, I am considering the radial distance from 0 to infinity. 0 to infinity the distance I am considering the radius. But angle I am considering from theta 1 the angle is suppose this theta 1 to this direction is supposed Theta 2. So, that is and this angle theta 1 and theta 2 I am considering and that means, I am considering this wedge, this I am considering.

And corresponding to this I am determining the feature, the feature is $f_{\theta_1 \theta_2}$ from the power spectrum. So, this ¿$f(u,v)$∨¿^{2}¿is the power spectrum. So, from the power spectrum, I am determining these 2 features, once is $f_{r_1 r_2}$and another one is $f_{\theta_1 \theta_2}$.

So, that means, we can see the behavior of the texture, the fineness or the coarseness and also we can see the direction of a particular texture pattern, that we can see from these 2 features, one is $f_{r_1 r_2}$, another one is $f_{\theta_1 \theta_2}$ So, this is about the autocorrelation function for representation of a particular texture.

(Refer Slide Time: 17:31)



The next one is very important, and that is the Gabor filter-based texture representation. That is I can extract texture features by using a Gabor filter. So, what is Gabor filter you can see a 2D Gabor function consists of a sinusoidal plane wave of a certain frequency and orientation

modulated by a Gaussian envelope. That means I am considering its 2D sinusoidal function. So, here you see $S_{\omega,\theta}(x, y)$that is the sinusoidal plane wave and I am considering the frequency, the frequency is omega and orientation is theta.

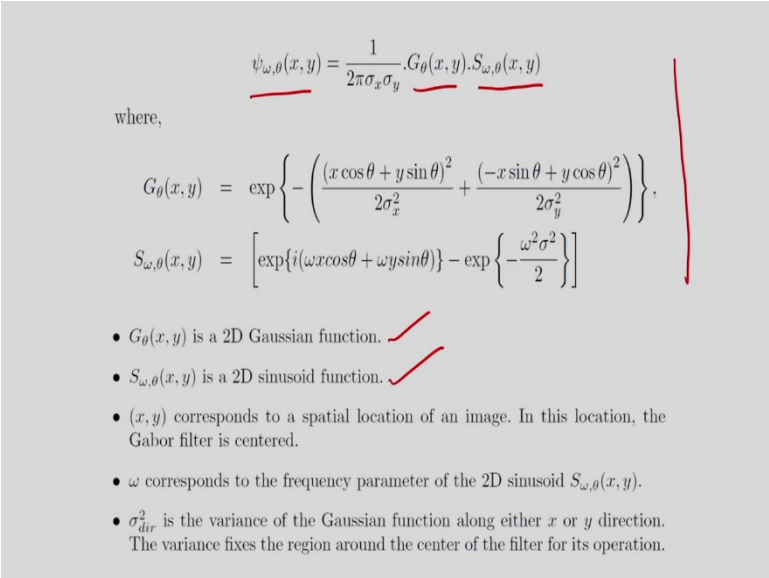
And that is modulated by a Gaussian function that Gaussian function is $G_{\theta}(x, y)$, that is the Gaussian function. And if I consider these 2 functions, one is the sinusoidal function, another one is the Gaussian function then I will be getting the Gabor function, the 2D Gabor function I am getting. And in this case you can see the $G_{\theta}(x, y)$ that is the Gaussian function, the 2D Gaussian function and in this case you can see the σ_x^2 that is the variance along the x direction and theσ_y^2, that is the variance along the y directions.

This actually represents the spread of the Gaussian function. And in this case S omega theta that is the sinusoidal plane wave you can see, so it has 2 components one is the imaginary part, another one is the real part. Here I am considering the imaginary is you can see. So it is a complex function and I am considering a sinusoidal plane wave of a certain frequency, the frequency is omega and the orientation is theta. This sinusoidal function is modulated by the Gaussian.

The 2D Gaussian function is used to control the spatial spread of the Gabor filter. The various parameters of the Gaussian determine the spread along the x direction and the y directions and also, we have done orientation parameters. So, in general, in general for the Gaussian filter, the $\sigma_x = ¿\ \sigma_y = \sigma$ that means that is a variance along the x direction is equal to variance along the y direction is equal to sigma is a constant.

In this case, the rotation parameter theta does not control the spread as the spread will be circular, if variances are equal. In this case, I am considering the sigma x is equal to sigma y is equal to sigma that means, the variance along the x direction is equal to variance along the y direction. And corresponding to this case, the rotation parameter Theta does not control the spread, as the spread will be circular, if the variances are equal. So, this is the definition of a Gabor filter. That means we are considering a sinusoidal plane wave of a certain frequency and orientation modulated by a Gaussian envelope.

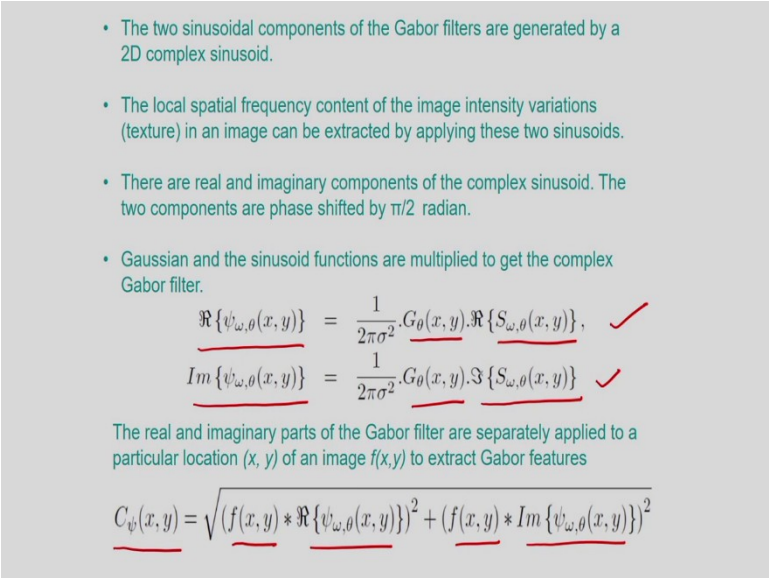
(Refer Slide Time: 20:57)



The same thing I am showing here in this slide also. I am defining the Gabor function, the 2D Gabor function and I am considering the sinusoidal function and also the Gaussian function. So, here you can see that $G_\theta(x,y)$ that is the Gaussian function the 2D Gaussian function, $S_{\omega,\theta}$, so I am considering the parameters the frequency parameter is omega orientation parameter is theta, that is the $S_{\omega,\theta}(x,y)$ is the 2D sinusoid function sinusoidal the sinusoidal function. XY corresponds to a spatial location of an image.

So, a spatial location of the image is considered and, in this location, the Gabor filter is centered. Omega corresponds to the frequency parameter of the 2D sinusoid. And Theta already I had mentioned that theta is the orientation parameters. And we have considered the variance along the x direction and the variance along the y directions. If this sigma x and sigma y is equal, then in this case, the rotation parameters does not control the spread as the spread will be circular if the variances are equal.

(Refer Slide Time: 22:16)



The 2 sinusoidal components of the Gabor filters are generated by a 2D complex sinusoid. The local spatial frequency content of the image intensity variation, that is a texture in an image can be extracted by applying these 2 sinusoids. And in this case, you can see there are real and imaginary components of the complex sinusoid. The 2 components are phase shifted by pi by 2. So, if you see the previous slide, you can see the sinusoid function that is a complex number and I have 2 components one is the cos theta another one is a sin Theta.

The phase difference between these is pi by 2. And in this case the Gaussian and the sinusoid function are multiplied to get the complex Gabor filter. So, here you see I am considering the ideal part of the sinusoidal function. So, R is the real and I am considering does $G_\theta(x, y)$ that is the Gaussian function, the 2D Gaussian function that is the real part of the sinusoidal function is modulated by the Gaussian and I am considering the real part of the Gabor filter.

The second one is I am considering the imaginary part of the sinusoidal function and this is modulated by the Gaussian and I am getting the imaginary part of the Gabor function. The real and the imaginary part of the Gabor filter are separately applied to a particular location, the location is suppose x and y of an image $F(x, y)$ to extract Gabor features. So here you see what I am doing. I am considering the images $f(x\,y)$ that is convolved with the real part of the Gabor filter.

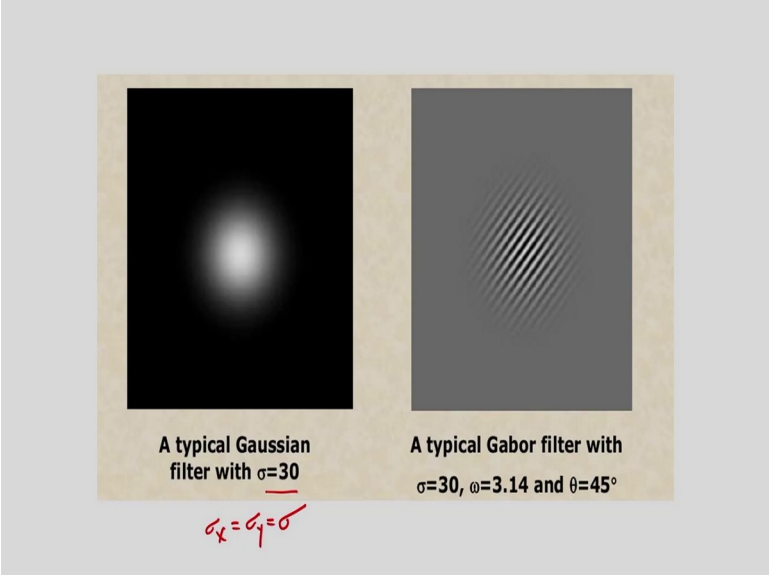
So this is the real part of the Gabor filter. And also you can see the next part, the second part, this is the image the image is $F(x, y)$ that is convolved with the imaginary part of the Gabor function and from this the real part and the imaginary part, I can determine that Gabor

coefficients, the Gabor features I can determine. So $C_{\psi}(x, y)$ that is nothing but the Gabor features. So, corresponding to a particular texture, I can determine the Gabor features.

So, the parameters are omega and the theta. Omega is the frequency and theta is the orientation. Also, another parameter the sigma x and sigma y. So, I will be getting the Gabor features for different orientations. So, you can see how we can extract the Gabor features by considering this Gabor filter. So, I have to consider the real part of the Gabor function. So, for this I am considering the real part of the sinusoidal function and also, I have to consider the imaginary part of the Gabor function, for this I am considering the imaginary part of a sinusoidal function.

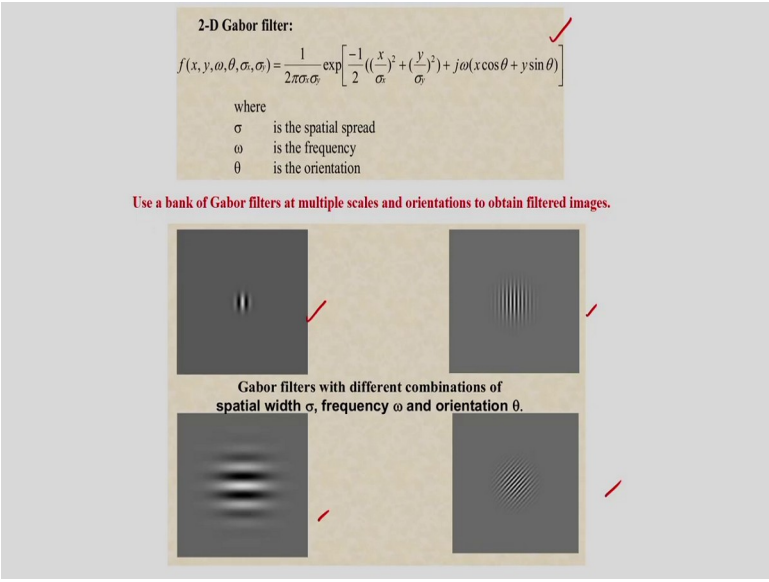
And already I had mentioned what are the parameters, the parameters are variance along the x direction, variance along the y direction, the orientation parameter theta and I can consider the frequency parameter omega.

(Refer Slide Time: 25:24)



So, in this case I have shown a typical Gaussian filter with sigma is equal to 30 and in this case I am considering the $\sigma_x = \sigma_y = \sigma$. And another case you can see a typical Gabor filter with sigma is equal to 30 and omega is equal to 3.14 and the orientation is 45 degree that I am considering.

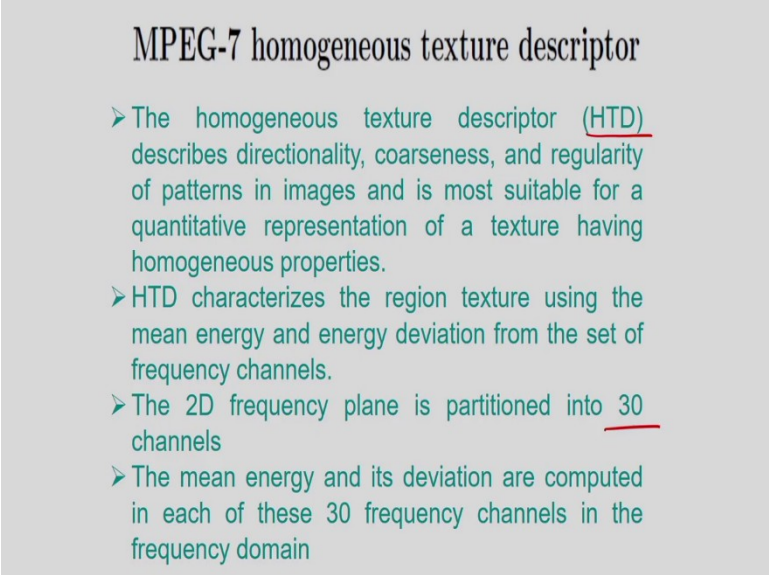
(Refer Slide Time: 25:47)



And here I have shown that 2D Gabor filter and this is the expression for the 2D Gabor filter and sigma is the spatial spread, omega is the frequency and theta is the orientation parameter. And in this case, you can see, use a bank of Gabor filters at multiple scales and orientations to obtain filtered image. Gabor filters with different combinations of spatial width, frequency omega and orientation theta.

So, you can see all these examples 1, 2, 3,4. So for this what I am considering the Gabor filters with different combinations of spatial width and frequency omega and the orientation also I am considering that is that theta. So, this is about the Gabor filter. So, by using the Gabor filter, we can extract texture features.

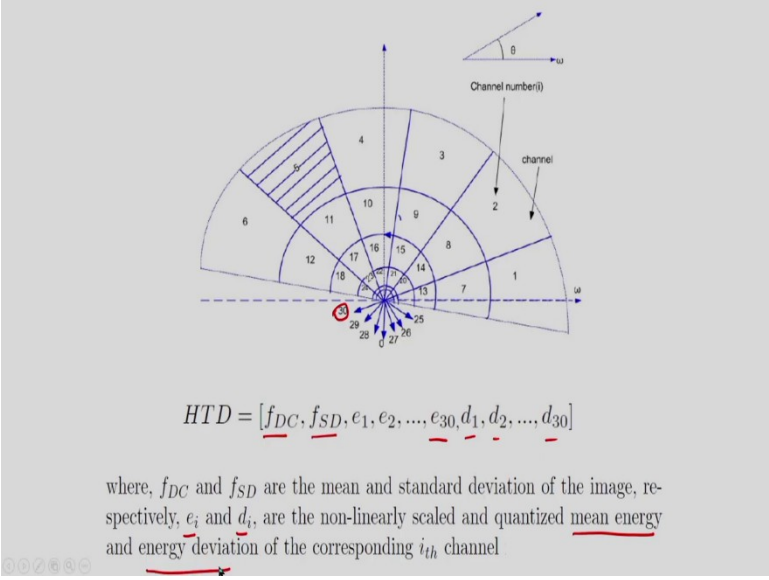
(Refer Slide Time: 26:38)



Another important technique is MPEG-7 homogeneous texture descriptors that is called HTD, the homogeneous texture descriptors. This HTD describes directionality and the coarseness and the regularity of a texture pattern. So, what it describes, the first one is the direction of a particular texture, the coarseness and the regularity of a texture pattern, it characterizes a particular texture. So, for this what I have to consider, for this the mean energy and the energy deviation from a set of frequency channels are considered.

So, I am repeating this. So, for HTD, the homogeneous texture descriptors, the mean energy and the energy deviation from a set of frequency channels are considered and 2D frequency plane is partitioned into 30 frequency channels. So, for this we have to consider 30 frequency channels. So, for this what we have to consider, the mean energy we have to compute and its deviation are computed in each of these 30 frequency channels in frequency domain.

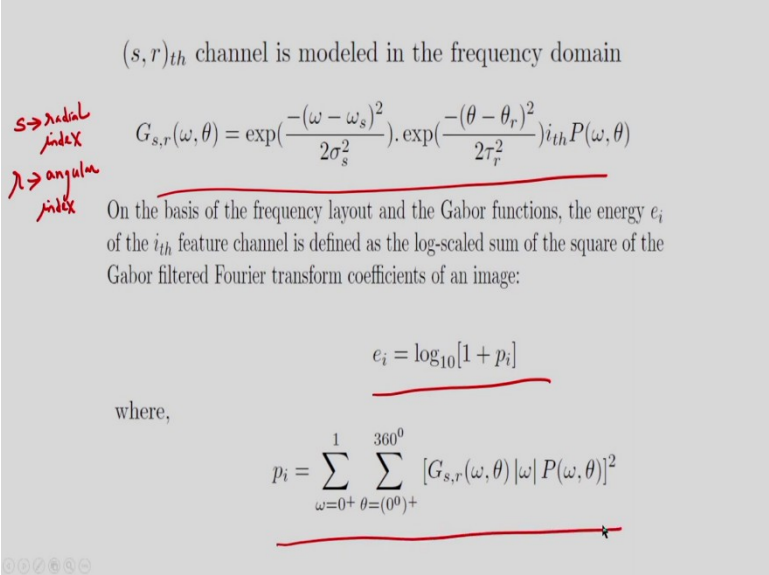
(Refer Slide Time: 27:48)



So, here I have shown the pictorial representation of the HTD, the homogeneous texture descriptors. And you can see we have considered 30 channels, the frequency channels we have considered. So, here what is HTD? So, here you can see HTD is the first component is F_{dc}. So, what is F_{dc} and what is F_{sd}. F_{dc} and F_{sd} are the mean and standard deviation of the image that is the definition of F_{dc} and F_{sd}, the mean and the standard deviation of the image respectively.

And I am considering e_1, e_2, e_3, up to e_{30} because we have 30 frequency channels and also we are considering the deviation is d_1, d_2 up to d_{30} for 30 channels. So, in this case the e_i and di are nonlinear victories scale and quantized mean Energy and Energy deviations of the corresponding ith channel. So, that means e_i corresponds to the mean energy and di corresponds to energy deviation of a particular channel.

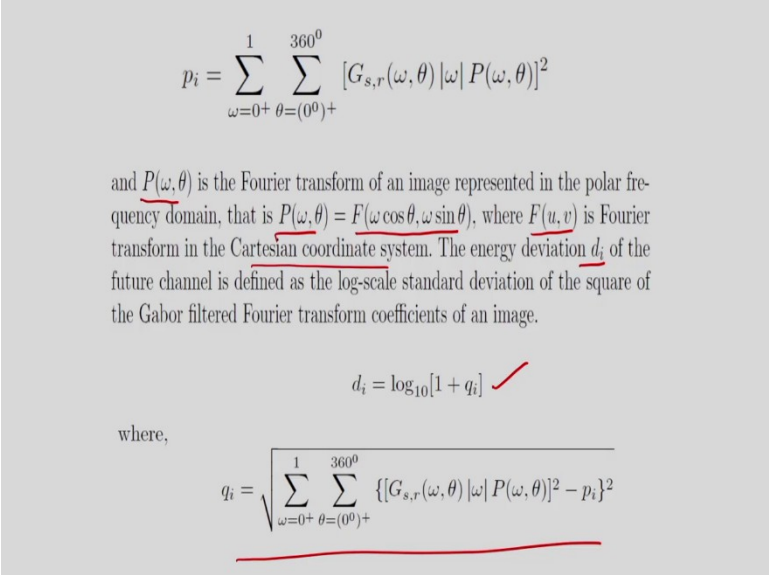
(Refer Slide Time: 29:09)



And also the individual channels or modeled by using Gabor functions. So, suppose let us consider channels index by s r. So, what is s here s is the radial index and r is the angular index. So, what I have to consider, the individual channels are modelled by using Gabor functions. So, I have 30 channels and the individual channels are model by using Gabor functions. And I am considering the channel index by s, r. s means the radial index and r means the angular index. So, s, r channel is model in the frequency domain like this.

So, this is G s r omega theta is nothing but I am considering 2 exponential functions you can see and P omega theta that I am considering. So, on the basis of the frequency layout and the Gabor functions, the energy ei of the ith feature channel is defined as the log scale sum of the squares of the Gabor filtered Fourier transform coefficients of an image. So, that means, what is ei, so ei is computed like this and you can see what is pi, pi it is determined from this expression that is the definition of pi.

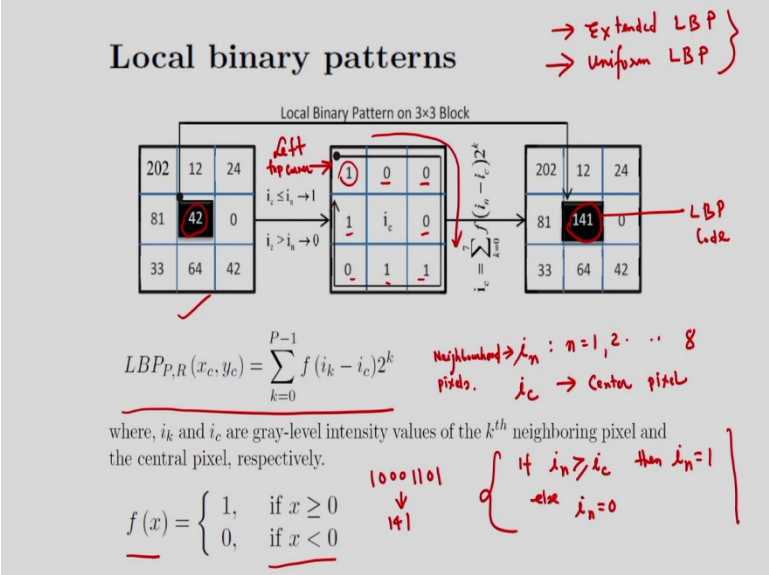
(Refer Slide Time: 30:46)



So, this is the pi. So, what is p omega theta in this expression if you see all this expression what is the $P(\omega,\theta)$. P omega theta is the Fourier transform of an image represented in the polar frequency domain. That is $P(\omega,\theta)$is equal to $f(\omega cos\,\theta, \omega\, sin\theta)$. And in this case f u, v is the Fourier transform, 2D Fourier transform of the image in the Cartesian coordinate system that I am considering. And also, I need to determine the energy deviation I have to determine.

In my last slide I have determined the mean energy. Now, I have to determine the energy deviation, that is the di for a particular channel. That is defined as the log scale standard deviation of the square of the Gabor filtered Fourier transform coefficients of an image. So, you can see I can compute di by using this expression and what is qi, the qi is obtained from this. This concept you can read from some papers on MPEG 7 homogeneous texture descriptors. So, that is by HTD, I can represent a particular texture, that is nothing but the texture descriptors.

(Refer Slide Time: 32:05)



And finally, I want to discuss another descriptor for texture, that is LBP, the local binary pattern. So, LBP is a well-known texture descriptor and that is widely used in many computer vision applications. So, for many computer vision applications, even the face recognition, facial expression recognition are the many applications the LBP is used. LBP is computationally simple and easy to apply and also, it is a nonparametric descriptor.

One important point is it can handle the monotonic illumination variation, that is one important point. So, it can handle the monotonic illumination variations. So, how to compute the LBP? So, corresponding to this image you can see here, that is I am considering the basic local binary pattern, because LBP has many variants, but in this case, I am considering only the basic LBP. So, in basic LBP each pixel of an image is compared with their 8 neighborhood pixels of a 3 by 3 block.

So, here you can see in this image, I am considering a 3 by 3 block and you can see the neighborhood of the pixel, the pixel is 42, the center pixel is 42 and the neighborhood pixels are 202, 12, 24, 0, 42, 64, 33 and 81, that means we have 8 neighborhoods. So, in that neighborhood, I can write like this i n i n, that is the neighborhood and in this case n is equal to 1, 2, so I have eight neighborhood pixels. Corresponding to the center pixel, the center pixel is suppose ic that is the center pixel. And after this what I have to do, I have to determine the LBP. So, how to determine the LBP.

So, for this if i n that is the neighborhood pixel is greater than equal to the center pixel, ic is the center pixel, this is the center pixel and in is the neighborhood pixel, this is the

neighborhood pixel. And if i n is greater than ic, then in this case, i n will be equal to 1 else i n is equal to 0. So this I am considering. So based on this you can see the 42 is compared 202. Since 202 is greater than 42, corresponding to this, I will be getting 1 here, this 1.

After this is the 12 is compared with 42, the 12 is Less than 42, so that is why I am considering it 0. Next, I am comparing 24 with 42, 24 is less than 42, so that is why I am considered it is 0. And like this I am getting these numbers, all these numbers I am getting. So, what will be the my number binary number 10001101. That means, I have to do the binarization. So, binarization is done by concatenating this in from the left top corner so, what is the left top corner is the left corner left top corner in the clockwise direction. So, clockwise direction is, this is the clockwise direction I am considering, this is the clockwise direction.

So, that means I am repeating this the binarization is done by concatenating this i n from the left top corner in the clockwise direction. So, corresponding to this my binary number will be 1000 if you see this example 1101 that is the case. And subsequently the corresponding decimal equivalent value I have to determine. So, corresponding to this what will be the decimal equivalent value? The decimal equivalent value will be 141, that 141 will be assigned to the center pixel, the center pixel is this is the central pixel.

So, this 141 will be assigned to the central pixel ic which is known as the LBP code and this step is repeated for all the pixels of the image. So, for all the pixels of the image I have to repeat this step and finally, the corresponding histogram of the LBP codes is considered as a local texture features of the original image. So, what finally I have to do. Finally, the corresponding histogram of the LBP codes we have to consider and the local texture features of the original image I can determine.

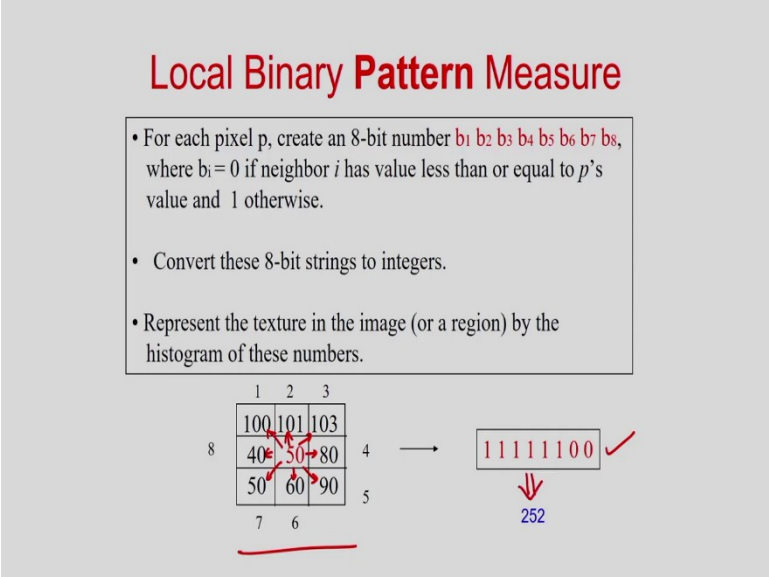
So, here you see corresponding to the center pixel I am getting the LBP code, the LBP code is 141. So, this is the LBP code for the center pixel. So, for all the pixels of the image, I have to do this and finally, the corresponding histogram of the LBP codes we have to consider and that is considered as the local texture features of the original image. So, I have to determine the histogram of the LBP codes.

The $LBP_{p\,r}$ corresponding to the central pixel I can determine and that will be the binary number and this binary number I can convert into the decimal number and what about this function, the function is F (x) the F (x) will be 1 if x is greater than equal to 0 otherwise it

will be 0, if x is less than 0. That means I am doing this case, that is I am comparing the center pixel with the neighborhood pixel and based on this I am getting the value of F (x).

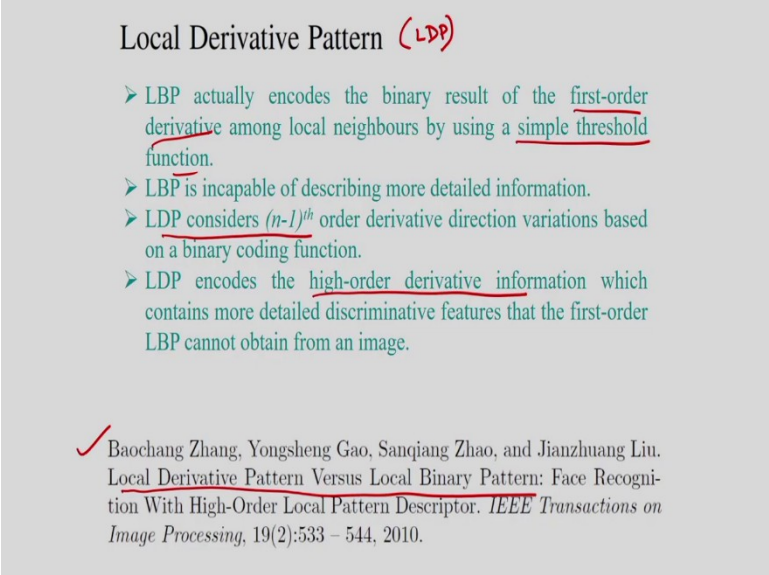
And from the F (x) I am getting this binary numbers and binary numbers I am converting into equivalent decimal representation and that is nothing but the LBP code. And there are several variants of the LBP. This is the basic LBP, like this I can give some example. One example is the extended LBP. Another LBP is the uniform LBP, that is very popular, uniform LBP is very popular. So, you can see all this concept from the research papers. So, what is extended LBP, what is uniform LBP. But the basic concept of the local binary pattern is this.

(Refer Slide Time: 38:58)



So, here also I have shown one example of the local binary pattern. Corresponding to this image you can see I am following the same procedure, just comparing the central pixel with the neighborhood pixels, just I am doing the comparisons like this. And based on this I am getting the binary code, the binary code will be 11111100 and the decimal equivalent will be 252, that is the LBP code.

(Refer Slide Time: 39:32)



And finally, there is another technique, I am not going in details about this local derivative pattern, that is called the LDP, the local derivative pattern. So, this is also very popular. If you want to see this concept, you can see this paper, the research paper, that is the local derivative pattern vs local binary pattern. So, in case of LBP, the LBP actually encodes the binary result of the first order derivative among local neighbors by using a simple threshold function. LBP is incapable of describing more detailed information.

So, that is why we consider the LDP, that means LDP considered n minus 1 th ordered derivative direction variations based on binary coding functions. An LDP encodes the higher order derivative, that is one also important point. In LDP, we consider a higher order derivative information, which contains more detailed discriminative features that the first order LDP cannot obtain from an image. Because in case of LBP, we consider the first order derivative.

And also, we considered a simple threshold function because I am comparing the center pixel with the neighborhood pixels and for this I am considering a threshold function. But in case of the LDP, we are considering the high order derivative we are considering and many directions I am considering, the directions like 0-degree, 45 degree like this different directions I am considering. And LDP encodes the higher order derivative information and that is why it contains more detailed discriminatory features in the LDP as compared to LBP.

So, for more details, you can see this paper on LDP, the local derivative pattern. So, in this class, I discussed some other texture features like autocorrelation function I have discussed

and after this one important representation that is a Gabor filter I have considered. So, how to extract the texture features by Gabor filter, that concept also I have discussed. This is very important. And finally, I discussed the concept of LBP, the local binary pattern. So, this is about the texture analysis. Let me stop here today. Thank you.