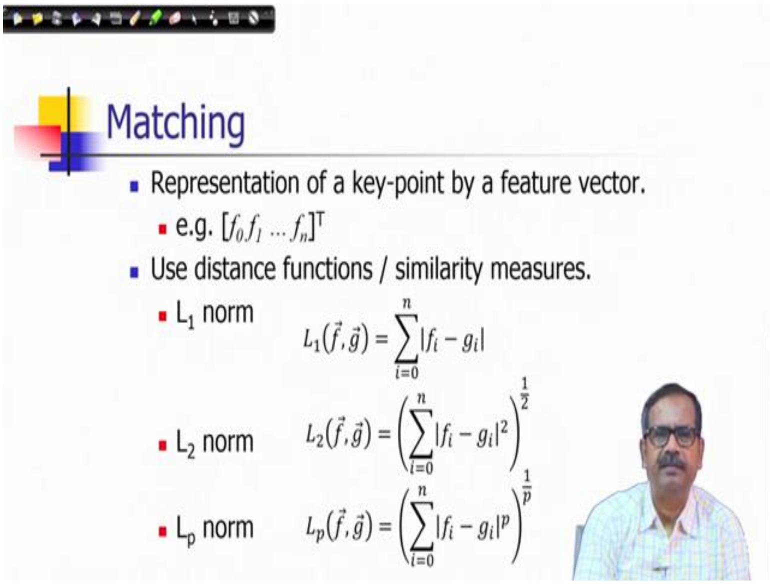
Computer Vision
Prof. Jayanta Mukhopadhyay
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 27
Feature Detection and Description Part – IV

We are discussing about techniques for Detecting Features and also Describing those feature points, so that we can use them later for matching with respect to various tasks.

(Refer Slide Time: 00:34)



So, let us consider briefly that what is meant by these operations matching and how you can compute it. In our next topic we will have much more elaboration on these particular computations, but to understand the motivation behind feature detection and descriptions let us also go through briefly about this competition.

So, one of the thing that we discuss that we can represent a key point by a feature vector. And suppose we have two images and they are related. They are of the same sign and we would like to find out the corresponding pair of points which are of the same scene point.

So, the key points which are of the same scene points needed to be identified. And they are the descriptors like feature vectors to be used, they are relevant in that case. And then you can use different distance functions or similarity measures to find out how close these points are. And accordingly we can take a decision.

So, for example a key point could be represented by feature vector. In this case we can see that the fields of the vectors are shown. And there are distance function for example, if I consider two feature vectors in the same way, so there are different norms which are defined mathematical in this way. So, this is a L_1 norm where you can see it is a sum of absolute differences between the fields of two feature vectors or components of two feature vectors.

$$L_1(\vec{f},\vec{g}) = \sum_{i=0}^{n} |f_i - g_i|$$

L_2 Norm which is Euclidian distance functions between two points or between two vectors.

$$L_2(\vec{f},\vec{g}) = (\sum_{i=0}^{n} |f_i - g_i|^2)^{1/2}$$

$$L_p(\vec{f},\vec{g}) = (\sum_{i=0}^{n} |f_i - g_i|^p)^{1/p}$$

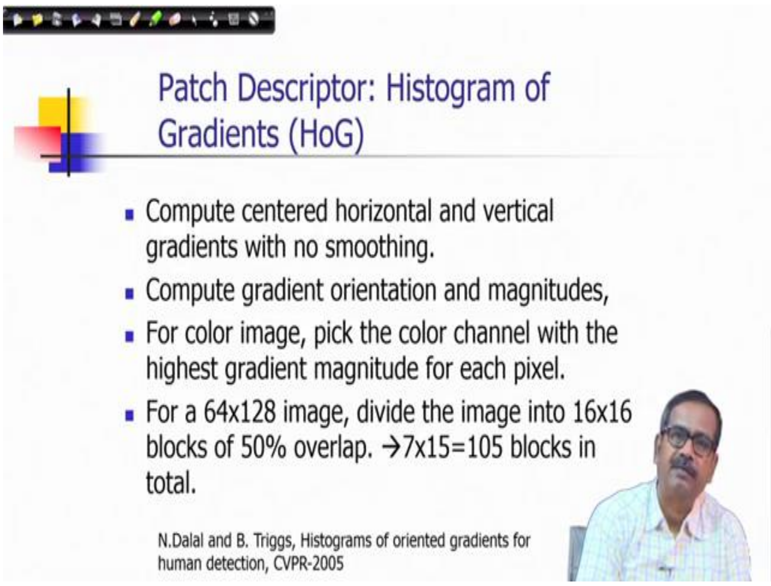
So, you can see that the special case of L_p norm isL_1 norm when p=1, and L_2norm when p=2. So, using this distance function, you can find out the proximities between two feature vectors into particular images. And which are more proximal we can assign those feature vector as the best. The closest one can be assign to a particular feature vector.

(Refer Slide Time: 03:37)

So, this is the kind of overview of how a matching could be performed using key point descriptors and then compute the corresponding points. So, let us consider different kinds of descriptors. It is not only the point descriptors sometimes to identify or to detect an object we require to describe their region instead of a point.

And we call those descriptors like patch descriptors or it could be a texture of a region. So it could be texture descriptors or some parts of image, or sub image. And they represent the whole content of that. So those are like global descriptors with respect to those regions. So, we will be discussing some of these techniques which we can derive this descriptors.

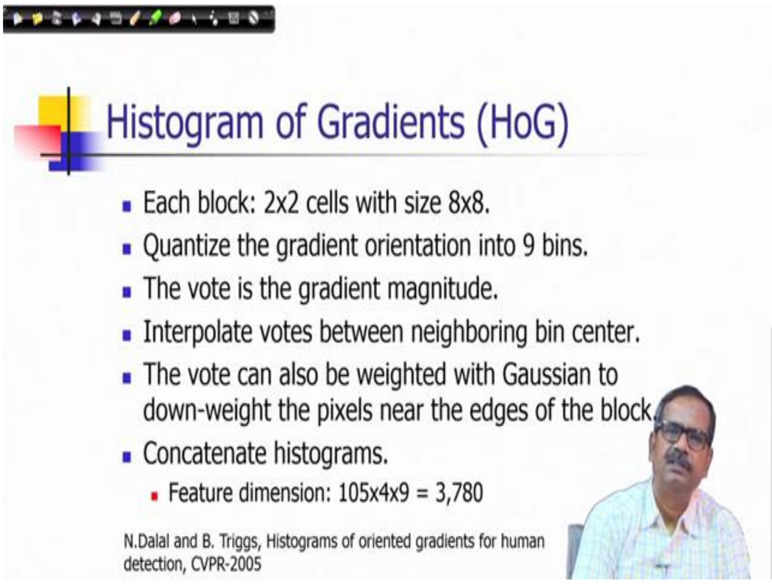
(Refer Slide Time: 04:30)



So, one of the very popular technique for patch descriptors is histogram of gradients representation. So, this particular method is proposed in 2005 as it is shown in the reference paper here and we can go through that paper to get the details. So, I will give you a brief overview of this particular technique. So, you are computing the horizontal and vertical gradients without any smoothing here.

So, you are collecting the gradient statistics like what we did in the serve descriptor etcetera, but here what we are doing? You are doing over a patch. So, we will be partitioning that patch and in each patch we will be computing this horizontal and vertical gradients. So that means, you have to compute gradient orientation and magnitudes and if you consider a colored image then you can pick that color channel where it is giving the highest gradient and use that directions in that case. So, typically say if you have a 64x128 image so, divide the image into 16x16 blocks of 50 percent overlap and then you have say 105 blocks in total.

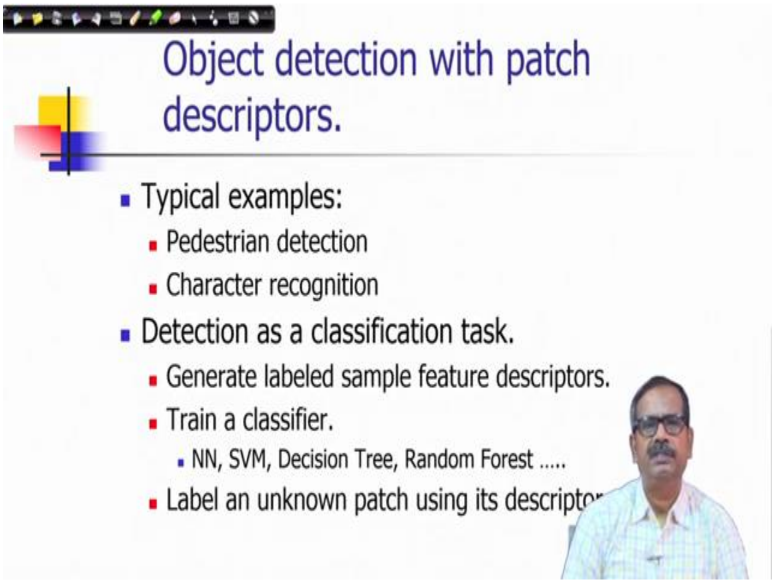
(Refer Slide Time: 05:59)



Then each block, partition into 2x2 cells with size 8x8 this is typically done in this papers. So, these statistics are taken from this paper. It has been found that this works well for certain object recognition and it has been found also in other techniques. Then you can quantize the gradient orientation into 9 bins and you can vote is the gradient magnitude. That means, you have each orientation as qualified or attributed by some of their gradient magnitudes along those directions.

So, it is like forming a histogram without counting each vector as a unit, we consider their magnitude as the weight of that particular directions. And accumulation of that magnitude gives the corresponding histograms weights of the histogram. So, interpolate votes between neighboring bin center, so you have to smooth the corresponding histograms and it could be weighted with Gaussian to down weight the pixels near the edges of the block. Then you concatenate at the histograms. So, it would give you a feature dimension as you have seen earlier there were 105 blocks and each block has 4 cells and each histogram has 9 bins. So, finally, you get a feature dimension of 3780 for a block of 64x128 patch.

(Refer Slide Time: 07:26)



So, how do you use this particular descriptor? So, this is a description and there are various examples of the application of these kind of descriptors. Like in that paper they have applied for pedestrian detection who are walking on a road.

So, their detection then character recognition in the text document. It has been found in various other applications where this hog descriptors are found to be useful with certain modifications. Usually these are descriptors and as you can see these problems are kind of a classification problem.

So, instead of matching using a distance function we can consider training a classifier. So, you can get a labeled sample feature descriptors and you can train a classifier, we can use distance function also considering that as representative samples of classes.

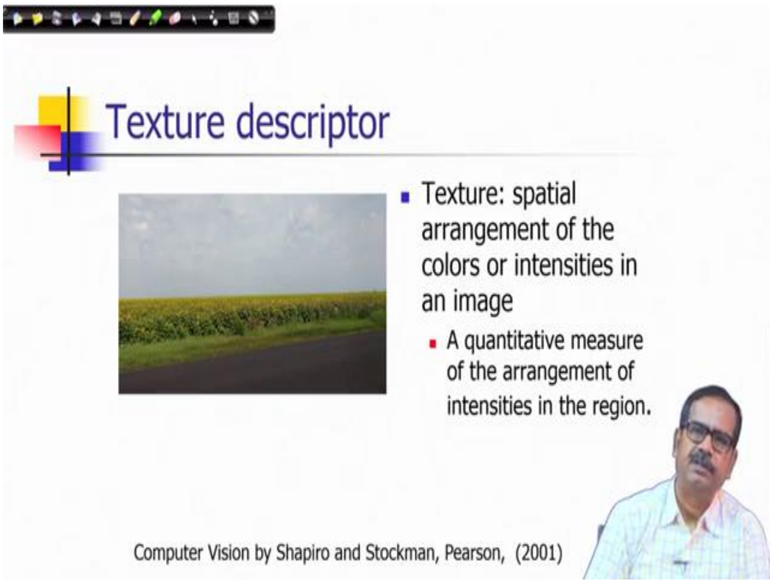
And then find out how close your unknown feature vector to those representative samples. But there are various techniques of machine learning which could be used for you know training this classifiers like support vector machine, decision tree, random forest and use them for that. So, by using those classifier later on you can once it is strained then you can label an unknown patch using this descriptor.

(Refer Slide Time: 09:00)

One of the operations that is often needed in this case is called non maximal suppression of this detection; which means that, sometimes when you detect a particular patch with certain class, even the neighboring patch also should give a high score. But as you understand that there exist only singularly one object should cover that particular radii and those matches are duplicate matches. And many of them will give you unnecessarily some false positive or redundant matching. The wise thing to do that is we ignore those matches.

So, that is the suppression; that means, as we did for key point detection. So, we consider the maximal response around a neighborhood here also we can consider maximal response around neighborhood patches and select that patch with locally maximal score. So, a greedy approach what has been discussed here is you have to select the best scoring window and then you suppress the windows that are too close to that selected window. And then again you carry out search on the remaining windows which are outside that region.

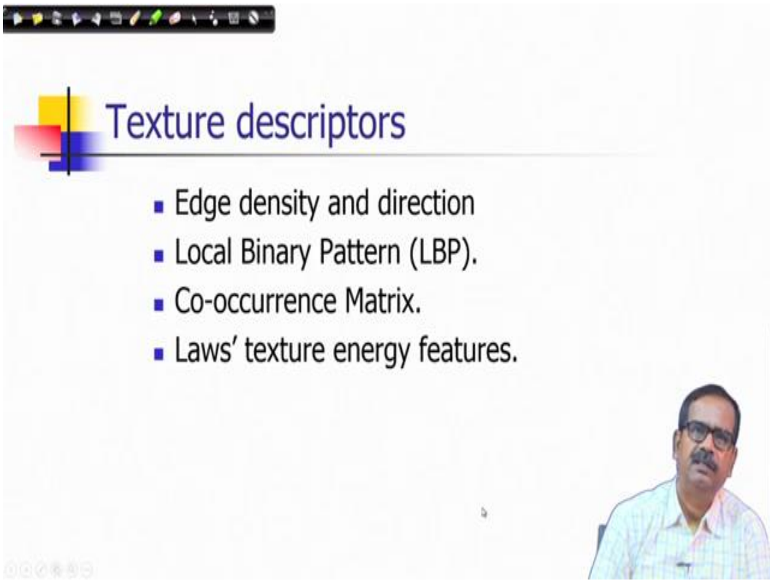
(Refer Slide Time: 10:24)



So, that is what a typical example of a patch descriptor and we will consider other kind of image descriptions particularly texture descript. And texture also describe a region and in this particular image you can see there are sunflower beds which has given a typical texture not only that the texture of sky, texture of road which are quite distinct.

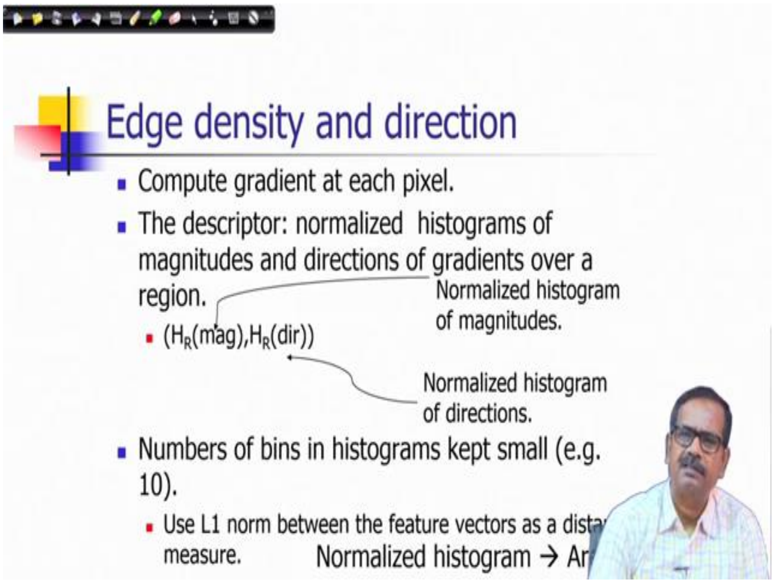
And those patches or those regions can be leveled with some of those classes. So, in brief, texture is a special arrangement of the colors or intensities in an image and we can define a quantitative measure to such arrangements and that measure itself distinctly identify a particular texture.

(Refer Slide Time: 11:19)



So, we will see how the texture descriptors are designed. So, there are various techniques for describing texture regions like these are certain techniques like edge density and direction, there are techniques using local binary pattern I will elaborate all these techniques. Then co-occurrence matrix and also 'Laws' texture energy features. So, 'Laws' is a researcher who has proposed long back this energy features and which have been found very effective in identifying textures.

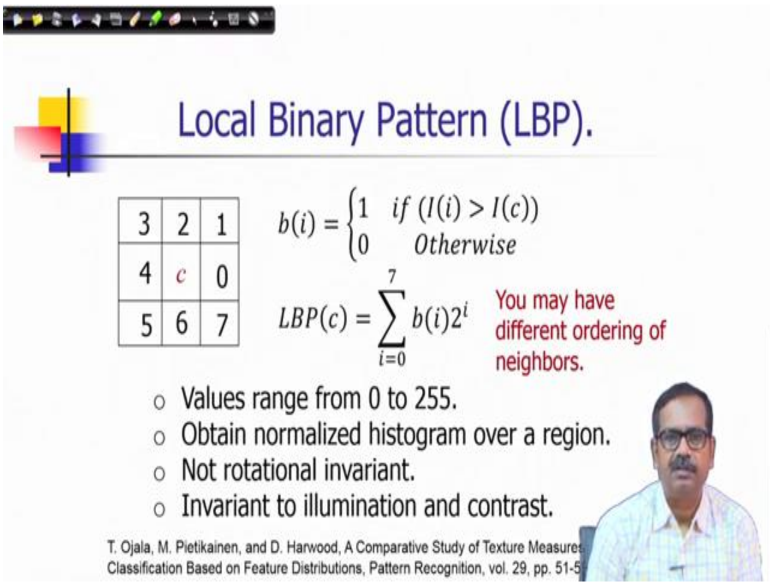
(Refer Slide Time: 11:55)



So, first the edge density and direction; In this method what we are doing is we are computing the gradient at each pixel and then we can get the histogram of that gradient and normalized histograms of magnitudes and directions of gradients over a region. So, you can get two histograms one for the magnitudes, one for the directions and you have to normalize it. So, in this case normalization means that you are normalizing with respect to the area of histogram so it is like giving a probability density function.

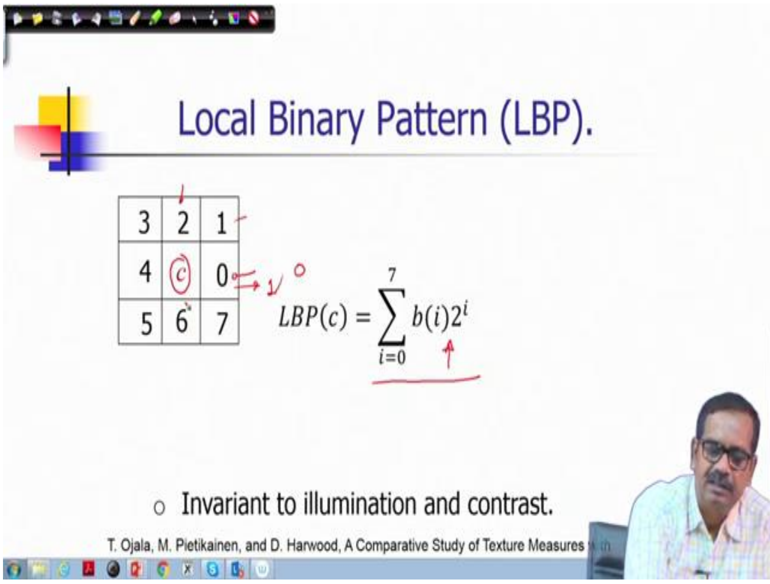
So, while making the area as one, so this is the representation that there are two histograms; one for magnitude, one for directions and they are normalized histogram of magnitudes, normalized histogram of directions as I mentioned. Typically number of bins in histogram they are kept small for example, 10 and then you can use some distance function like L1 norm between feature vectors to find out the levels of a texture. So, if you have a library of textures you can use this distance function, we will elaborate again this kind of matching letter.

(Refer Slide Time: 13:21)



This is another method by which textures are represented and this is a very popular technique called local binary pattern. So, let me define this particular feature.

(Refer Slide Time: 13:50)



As it has been shown that in this image you consider a central pixel and these are the positions which are numbered here. So, these are the neighboring pixels of the pixel 'c', but again they have given some positional number ordinal numbers.

$$\begin{matrix} 3 & 2 & 1 \\ 4 & c & 0 \\ 5 & 6 & 7 \end{matrix}$$

So, what we are trying to find out that we are comparing the pixel values between these two locations between the central pixel and the pixel at the locations.

$$LBP(c) = \sum_{i=0}^{7} b(i)2^i$$

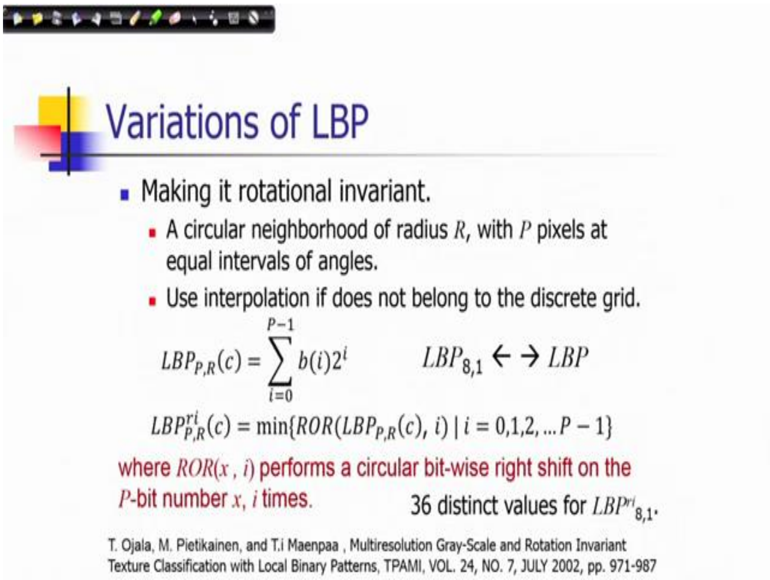
If the pixel value is greater than 'c', we will have a value 1 otherwise if it is smaller we will have a value 0, which means all these values will be a binary value. And that is a local binary pattern and which can be represented in an aggregated form by simply the value of that binary string. So, which is represented here by the above equation. It is simply the decimal value of that binary string.

So, that would describe this feature itself that binary value itself will describe the features and as you understand this will range from 0 to 255 in this particular case. So, this is the local binary pattern for an image and it can be shown it is invariant elimination and contrast this particular description local descriptions.

However, for a texture region what we need to do that this is the definition of b(i) as I mentioned that it could be either 1 or 0 depending upon whether the pixel at their location ‘c’ is greater than the central pixel or not. This is the typical example you may have different ordering of neighbors in some other work and accordingly the value would also changed. But the values would range from 0 to 255 and then what you do, you obtain a normalized histogram over a region.

So, you compute these value at each pixels then collect the statistics over that region in the form of a histogram and normalize that histogram. You should note that this is not rotational invariant. This description is not rotational invariant.

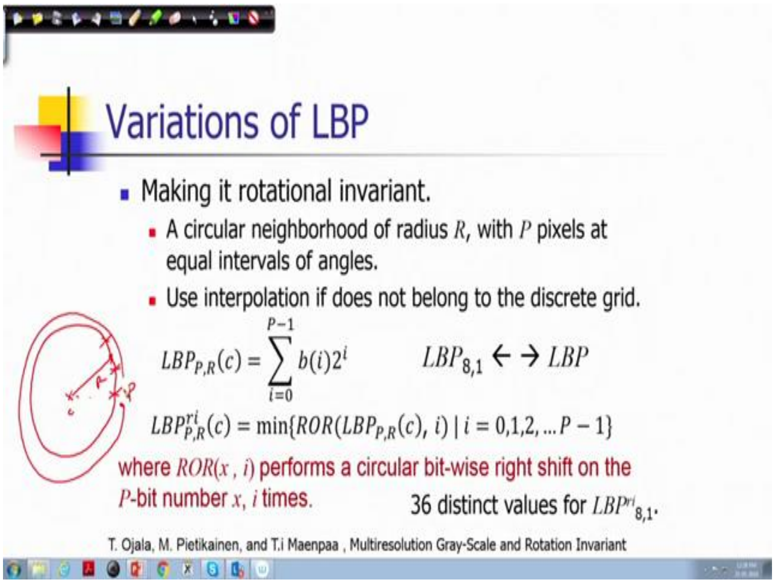
(Refer Slide Time: 16:04)



There are different variations of local binary pattern, so that you can make it rotational invariant. And in fact one of the reference paper has been cited here that is multi resolutions gray scale and rotational invariant texture classification with local binary pattern, it was published in a transactions of pattern analysis and machine intelligence in 2002.

So, I will be briefly describing this particular method as you can go through this paper and get the details. So, how do you make it rotational invariant ?. You consider a circular neighborhood of radius R with P pixels at equal intervals of angles. So, instead of having a fix 3x3 neighborhood let us extend this definition.

(Refer Slide Time: 17:09)



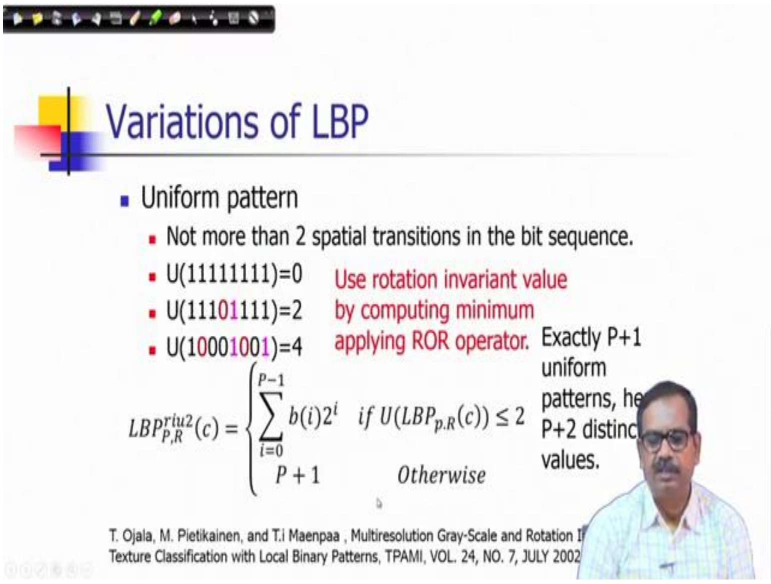
You are selecting P pixels in that neighborhood which means suppose I have a central pixel ‘c’. And then any neighborhood any radius if I draw a circle. And then at equal interval I will be selecting P pixels out of this, so the total number of pixels will be P. And then you consider the local binary patterns at those positions.

So, suppose since it is a discrete grid, if the pixel does not coincide with that particular location because of discrete grid. So, you can interpolate in a grid it is represented by 2 pixel locations in this continuous form, so, you have to interpolate them. The thing is that after those values you can get a local binary pattern value, and those are characterized by these parameters: number of pixels ‘P’ that has been taken over that circular neighbor and also the radius.

So, typically the ordinary local binary pattern what we define in the previous case that would be the same when the radius is kept as 1 and where the number of pixel is kept as 8. Now, the rotational invariant LBP is defined that you consider the rotation of this binary string on those locations P that you perform rotational right rotation operations for example. And in every rotation you compute whatever the value decimal value you get and the minimum one you will be choosing.

So, finally, as you can see when you if rotate the image, the same value will be there because you are always taking the minimum out of all possible rotations. So, that is how you make it rotational invariant, so if you are performing a circular bitwise right shift operations for this rotation. And it has been shown that there are 36 distinct values if I consider a rotation invariant representation when radius is 1 and P value is 8.

(Refer Slide Time: 19:23)



There is another variation instead of choosing all sorts of local binary pattern, in this case we will be considering only those pattern which we will considered uniform. And rest of them would be put in the same class or in a separate class

$$LBP_{P,R}^{riv2}(c) = \begin{Bmatrix} \sum_{i=0}^{P-1} b(i)2^i,\ if\ U\left(LBP_{P,R}(c)\right) \leq 2 \\ P+1, otherwise \end{Bmatrix}$$

So, which means there would be exactly you know P+1 class is when we define the uniform pattern in this way that there, it is the pattern where there is not more than two special transitions in the bit sequence. And so, it is you know the number of transition is 0, so it is an uniform pattern.

$$U(11111111) = 0$$

Consider, in this case there is a special transitions say 1 to 0 and 0 to 1 only that is number is 2, so this is also an uniform pattern.

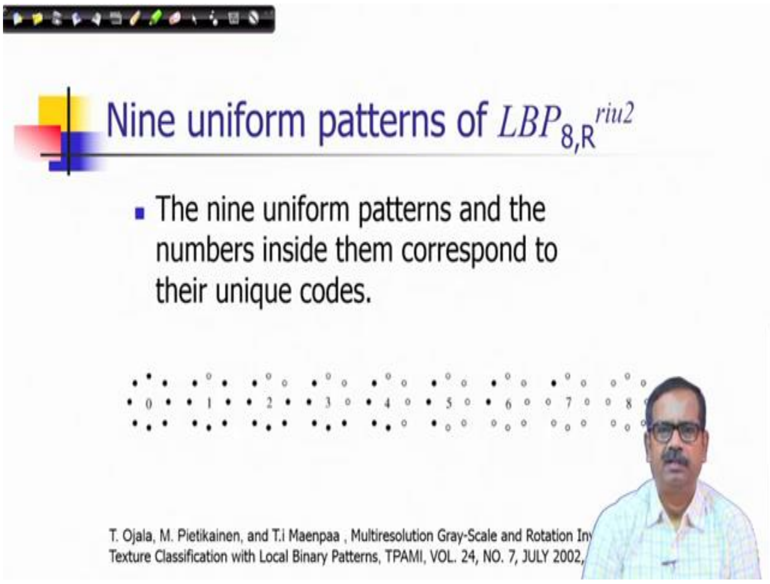
$$U(11101111) = 2$$

Whereas, you consider the third example and you can find the transitions in the beginning there is 1 to 0 then in the middle there is 0 to 1 and 1 to 0 and at the end there is 0 to 1. So, there are four transitions, so this is not a uniform pattern.

$$U(10001001) = 4$$

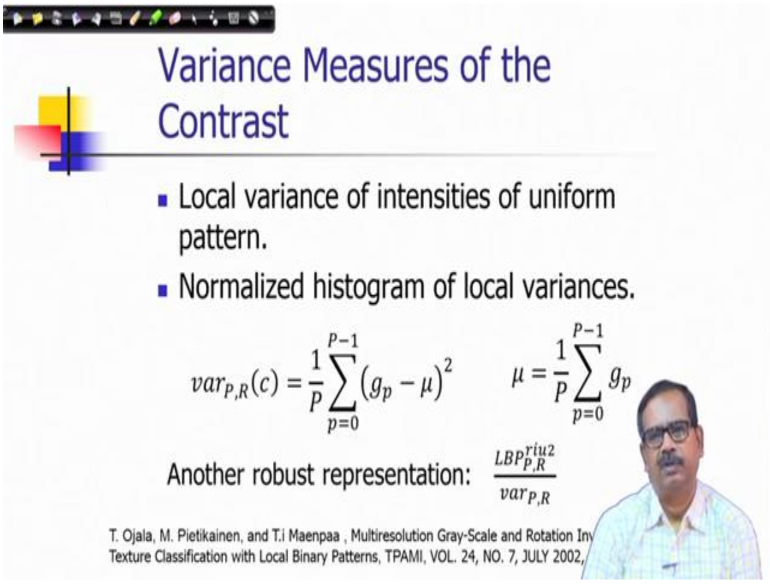
So, now, how do you define the LBP in this case rotational invariant that you are considering only those pattern which is uniform you are taking their rotation invariant value? So, which means you perform right shift operations take the minimum value; right shift rotations to the minimum value otherwise you put them in a separate class in a separate value as P+1. So, in this way there could be at most P+2 distinct values. So, you have to use rotation invariant range value by computing minimum applying ROR operator.

(Refer Slide Time: 21:13)



So, it has been found there are 9 uniform patterns in the number inside them that correspond to their unique codes, so these are certain examples which have been shown in that paper itself. So, these are the uniform patterns, rest of them will be non uniform and you get 9+1, so in total there would be 10 distinct classes. So, if I take the histogram of this local binary patterns there would be only 10 bins in this case.

(Refer Slide Time: 21:47)



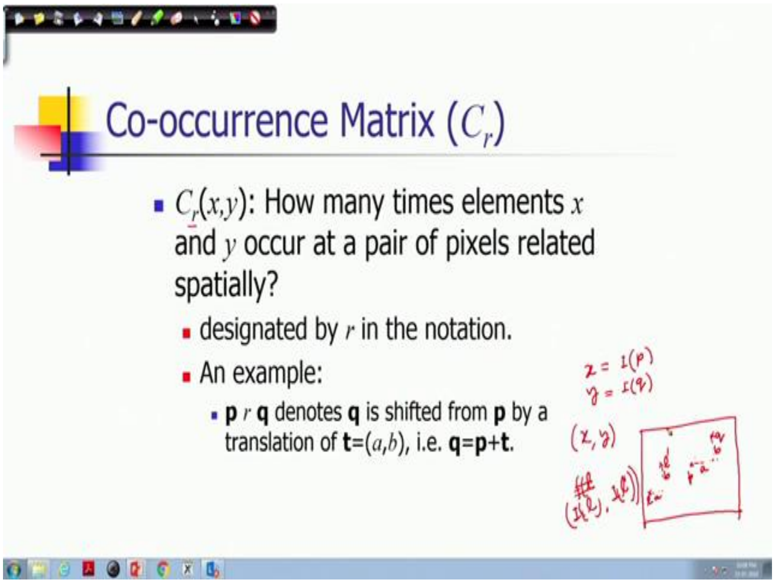
$$var_{P,R}(c) = \frac{1}{P}\sum_{p=0}^{P-1}(g_p - \mu)^2 \quad \mu = \frac{1}{P}\sum_{p=0}^{P-1} g_p$$

And also this could be further augmented by other rotational invariant measures or other measures which would make it intensity invariant. So, local variance of intensities can be considered for uniform pattern, so this is the definition of variance over those values.

So, it is only those pixel intensities for uniform patterns those are the positions and you are taking the variance. So, only for uniform pattern you are considering the local variance and this is the main thing. And then you got the normalized histogram of local variances also, so this is another feature.

So, you can get normalized histogram of LBP or you can get normalized histogram of local variances and in fact, you can get another robust representations when you take this ratios. So, these are different representations by which a region can be described either you can describe it as an histogram of rotational invariant local binary patterns as we defined or histogram of local variances as it is defined here or even histogram of you know ratios of this two quantities.

(Refer Slide Time: 23:15)



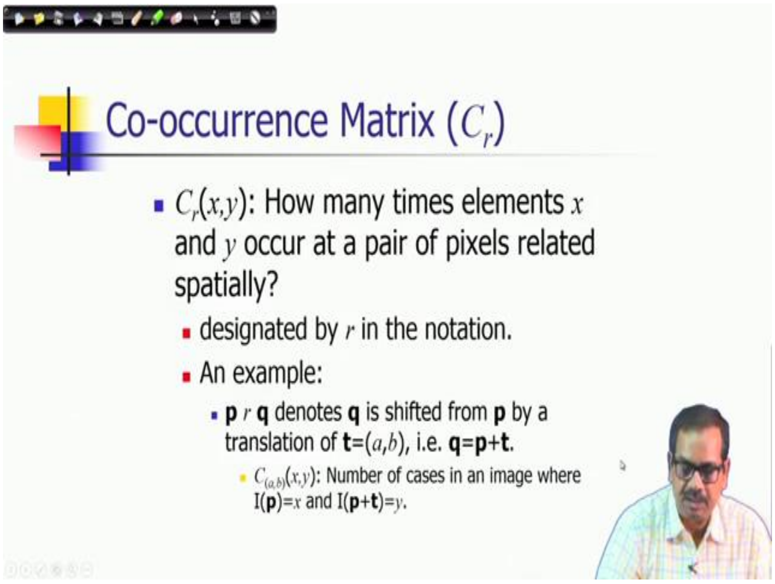
So, let us understand what is meant by a co occurrence matrix as it has been already shown in the slide. That it is a matrix where every element of that matrix defines or contains that how many times the indexes of those elements. That means, x and y those are the pixel values how many times they occurred together with respect to certain spatial relationships.

So, in this notation you note here that we have use to a subscript ‘r’, here ‘r’ is denoting that pixel location. So, for an example we consider there are two pixel ‘p’ and ‘q’ which is shifted by a translation vector a b, then the pixel values; that means, if their intensity values. Then ‘x’ is intensity values at location ‘p’ and ‘y’ is intensity value at location ‘q’, and an occurrence of (x, y) that would be considered that would be counted in this case.

So, now throughout the image you find out any pixel location, so let me explain once again suppose we have an image and suppose this is ‘p’ and this is ‘q’ and there is a horizontal and vertical shift a and b there is a translation term, so this is how the p and q is located. So, if I take another pixel say [laughter], then once again its corresponding pixel L prime would be here which is also shifted by the same amount. So, in that case we will consider what is the intensity values at l and l prime and this pairing will be considered that would provide you the corresponding you know statistics of how many times this pairs have occurred. So it is a frequency distribution of all such pairs.

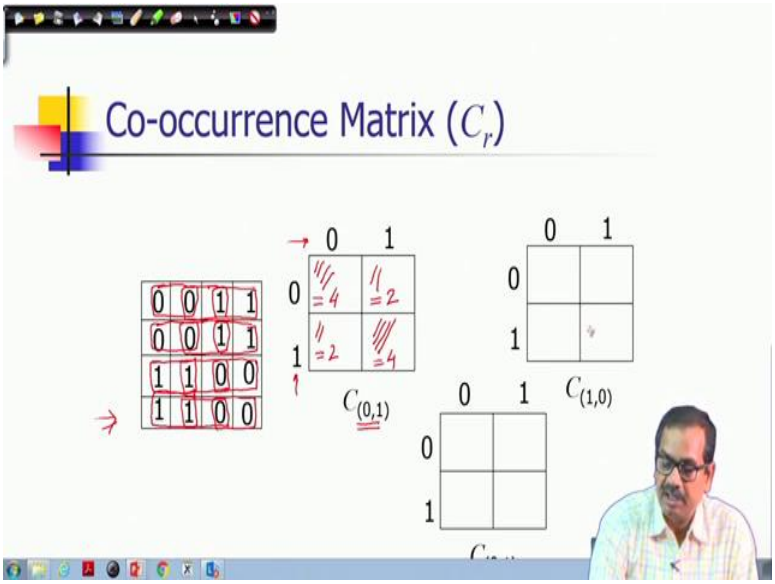
So, suppose in an image there are 256 levels, so you will get 256 square such pairs and for each combination you have to find out how many times that combination occurs throughout the image. So, when you have a matrix in this form; that means, (i, j) th element of that matrix denotes the pixel value i and pixel value j in corresponding locations with satisfying this special relationships between those locations throughout the images how many times that has occurred. So, this will give you the frequency distribution that is how the co occurrence matrix is defined.

(Refer Slide Time: 26:48)



So, this is also explained here, so if I consider a relationship ‘r’ where there is a translation of a and b as we denoted. So, number of cases in an image where I p equal to x and I p plus t which is a translation vector a b that is equals y.

(Refer Slide Time: 27:20)



So, let us workout a simple case from where we will be able to understand this particular relationship in a better way. Considered a small image a 4x4 image and the image has only two levels which is 0 and 1.

So, if I consider the special relationships of 0 and 1 then if I define a particular location for example, if this is my location p then its corresponding q location as per the relations would be this one so we will be considering the pairing value 1 0. So, this denotes the value of the location 'p' or the originating locations or the reference location if I say and this is the other second locations in that couplet, so this denotes those values of second location.

So, if I say it is 1 0, so 1 and 0, so there will be a count for this one. So, in this way let us see what are the locations where 1 0 has occurred. So, this is one, this is another instance where which satisfies this relationship, so we can take this then I think that is all. Let us consider the other cases, say this is 0 0, so we will take this, this is also 0 0 we will take this we will go on doing this see this is 0 1.

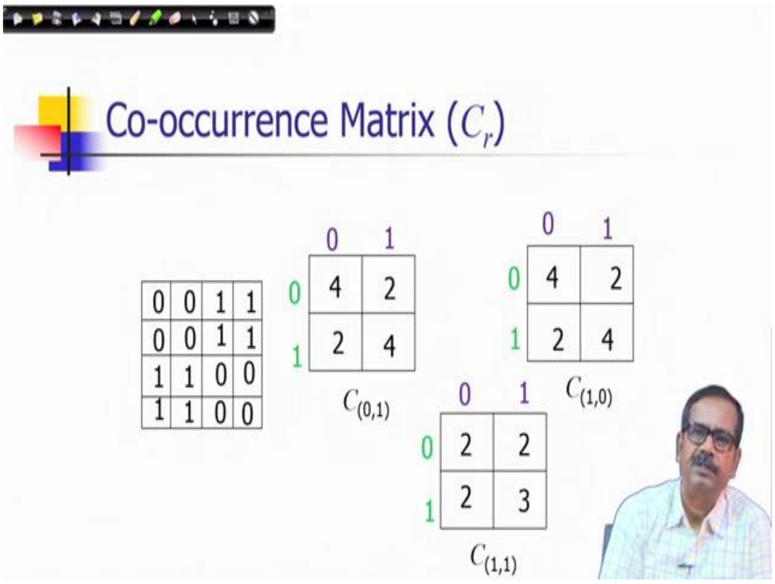
So, 0 1 this one this is 1 1, so 1 1 this one we have already considered this location this is 0 1. So, 0 and 1, this one this is 1 1 this one similarly this value is 1 1. Now this has been already considered see this is 0 0. So, 0 0 will be this one this is 1 1, 1 1 this one oh not sorry this 1 is 1 1, so here is 1 1 earlier then this is 0 0.

So, now, you count how many times it has occurred, this has occurred 4 times; these has occurred 2 times, 2 times, 4 times. So C (0, 1) this co occurrence matrix will look like this.

$$\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$$

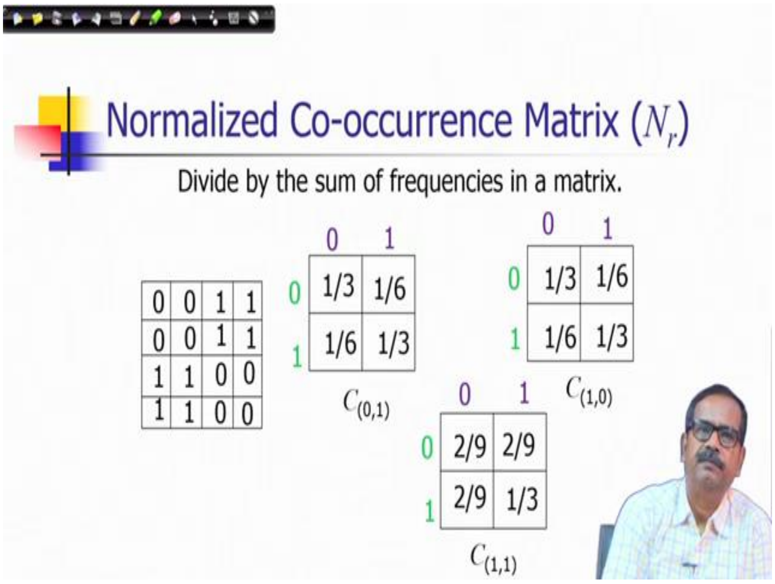
So, similarly you can compute to other frequency distributions and which will give you these are the values that you can find out that.

(Refer Slide Time: 30:52)



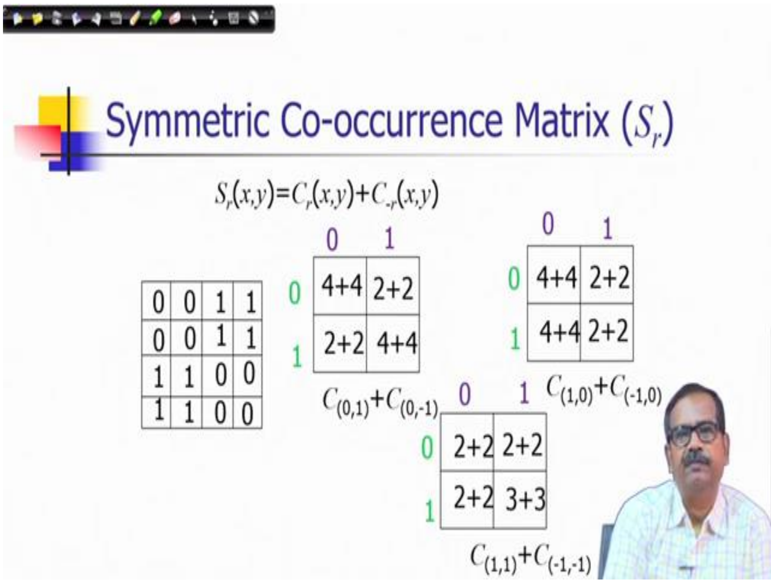
So, every different positional relation we will have a different co-occurrence matrix. So, depending upon the definition of the coupled locations of the pixel you get a different co occurrence matrix. And this set of co occurrence matrices they represent a texture feature.

(Refer Slide Time: 31:22)



You can normalize this co occurrence matrix by dividing them by their sum of frequencies in a matrix.

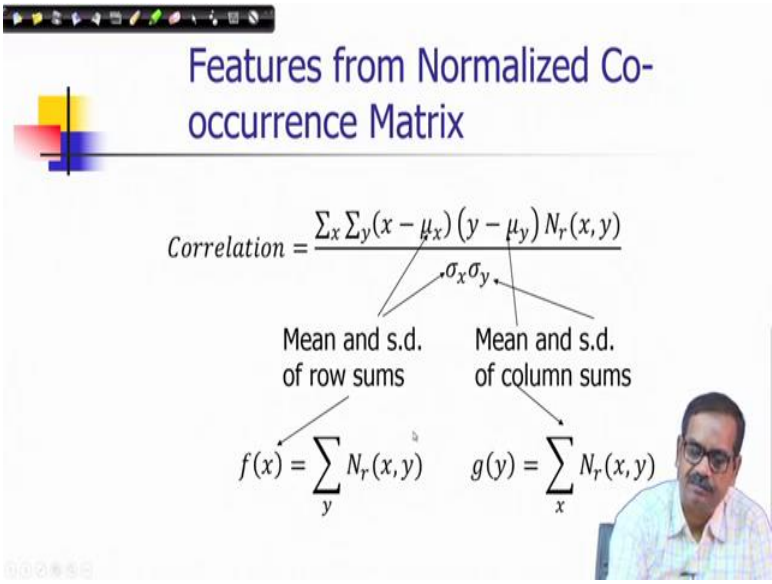
(Refer Slide Time: 31:30)



$$S_r(x,y) = C_r(x,y) + C_{-r}(x,y)$$

Or you can also modify the definition by using symmetric co-occurrence matrix; that means, you can consider the directions in the opposite directions also say shift (a, b) again minus a-b. Like 0 1 0 minus 1 and find out the counts at for each case and then aggregation of those counts will give you a co-occurrence matrix for each element of that matrix.

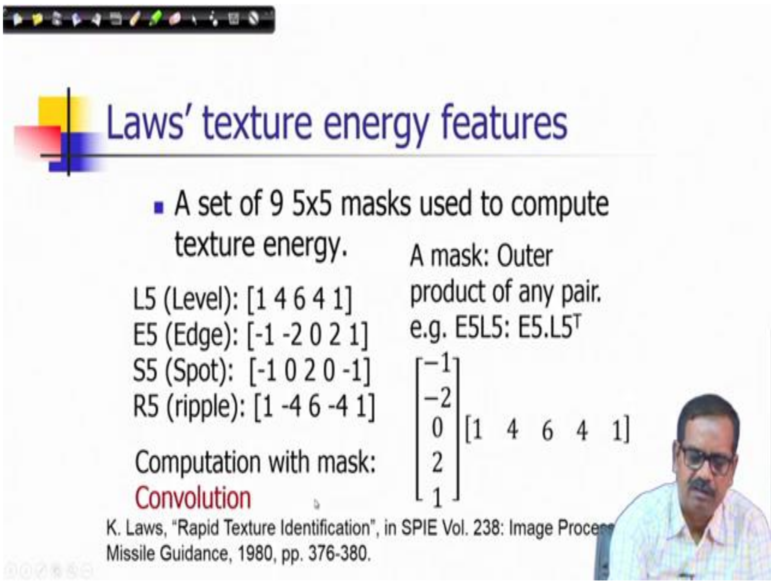
(Refer Slide Time: 32:00)



$$Correlation = \frac{\sum_x \sum_y (x - \mu_x)(y - \mu_y) N_r(x, y)}{\sigma_x \sigma_y}$$

So, from this co-occurrence matrix you can define different features and those features can describe a region. So, co-occurrence matrix is not just describing region; from the co occurrence matrix different measures can be defined like correlation measure. So, we can see that this is a normalized correlation measure which means it is weighted by the corresponding you know value of the co-occurrence matrix.

(Refer Slide Time: 33:22)



So, in this way you can define many other features and you can represent them by a feature vector. The last technique which I will be discussing for texture representation is Laws' texture energy features. I have already cited the paper which you can go through. It is a very old paper in 1980, but these features are found to be very effective.

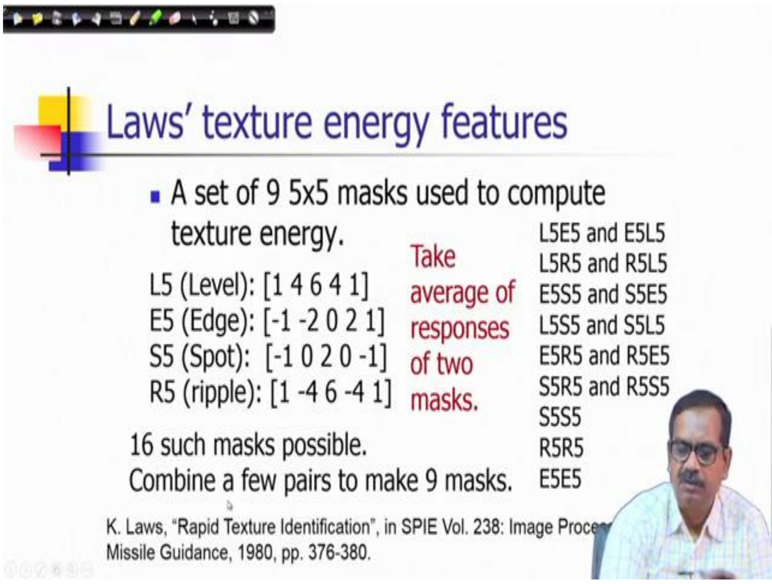
So, how this features are computed there are set of 9, 5x5 masks which is to compute this texture image, so let us define how this 9 masks derive. In the base level you can see there are 4 one dimensional filters. And their element is there are 5 such elements of the filters that is why this abbreviations L5, E5, S5, R5 those are denoted here and each one has its purpose of competitions.

So, if you have a one dimensional filtering you can see that L5 it is just performing an weighted small thing. Where is E5 it is performing a gradient computations along the x direction [-1 -2 0 2 1] in one dimension it is computing the gradient directions. Say spot S5 is computing spot which means you know the central pixel is the higher one and then you are taking the differences it is a center surround model. So, you are masking the surroundings surrounding excitations or surrounding responses are subtracted from the central response that the, and R 5 is a ripple it is also given by this particular mask.

Now, this is for one dimensional computation's, but when you do it for two dimension then you perform outer product of any pair, so outer product that it is a product of this matrices. So, if I represent them in the form of a column vector say all these vectors you can represent as a column vector, then outer product is the column vector into a row vector.

Suppose the outer product of E5, L5 we will give you E5L5 transpose. Say this is one example of outer product of E5 and L5 transpose and this will give you a mask. And then you perform the convolution with the corresponding texture. So, this is typically one example of a 5x5 mass which computes a texture response.

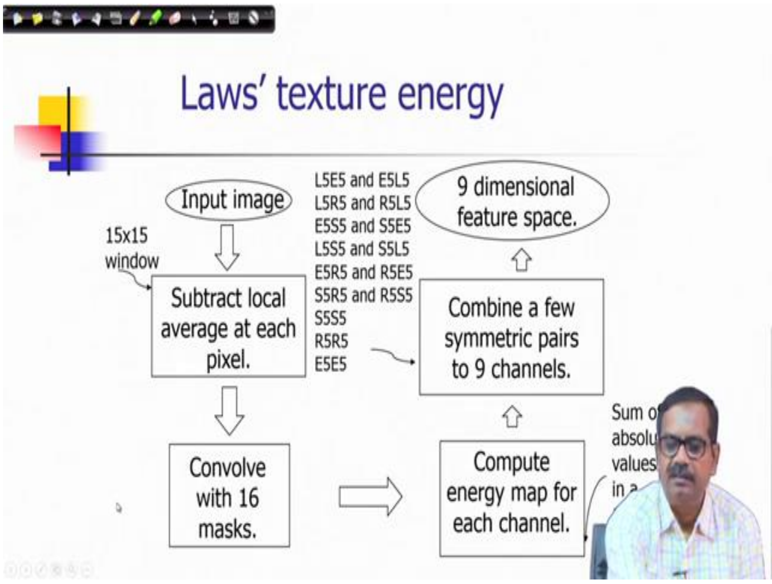
(Refer Slide Time: 36:12)



And from there we compute the energy, which means that has to be sum of square of all these responses. We will come to that details. So, this is a set of 9 5x5 masks so, 16 such masks are possible because, there are four. There are four pairing possible but, we combine if you pair to make them 9 masks. So, the list is given here we can see that some of them are trying to provide you the symmetric measures like L 5, E 5, E 5, L 5 they are combined.

So, in this way we can find that there are 6 into 2; 12 masks they are combined in each pair and they make it 6 and then there are masks S5, S5, R5, R5, E5, E5 they produce 3 masks. So, they take average of responses of 2 masks that is how you combine them.

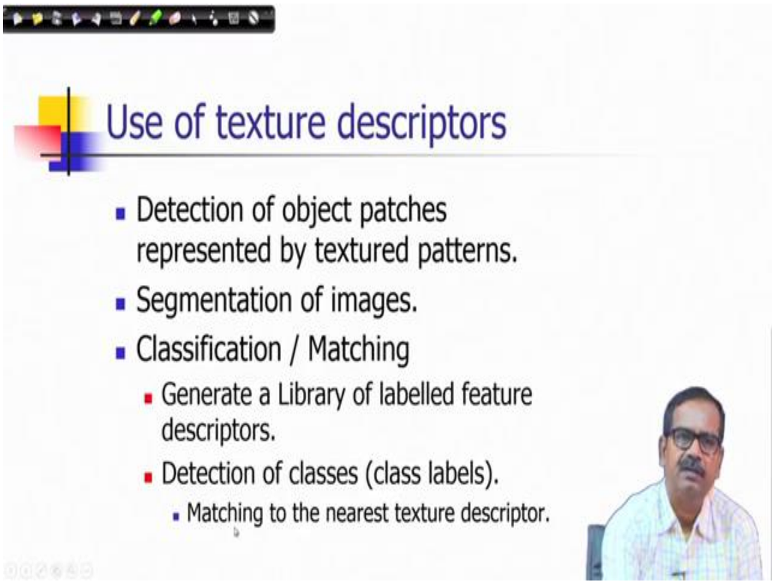
(Refer Slide Time: 37:13)



So, finally, as I representation these are the steps that you have an input image, then you can subtract the local average at each pixel convolve with 16 masks computer energy map for each channel then combine a few symmetry pairs to 9 channels.

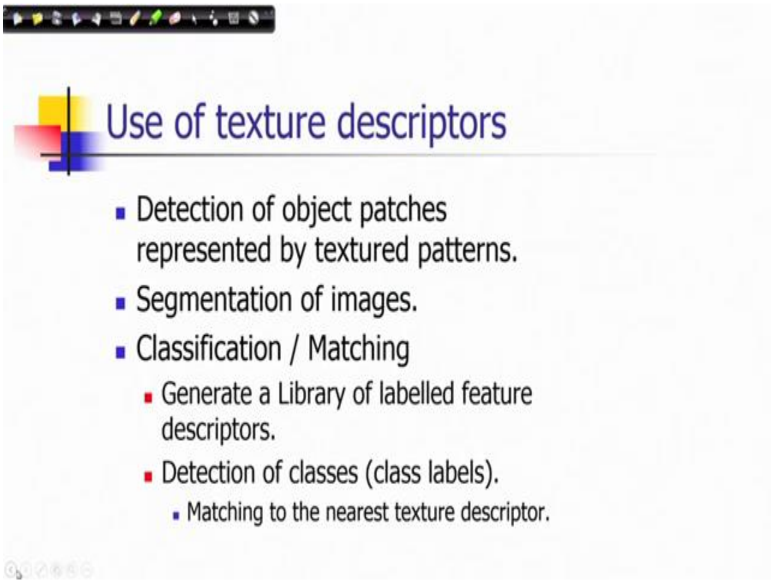
So, you have a nine dimensional feature space; that means, every pixel in the textured region has a 9 dimensional vector representation. So, this is a 15x15 window and this is a sum of absolute values in a 15x15 window, and you know these are the masks which are combined.

(Refer Slide Time: 37:49)



Some of the use of texture descriptors is like detection of object patches. It is represented by textured patterns, then segmentation of images and classification or matching. Usually, the problem is considered as a classification problem as the object the detections with using the region descriptors. Here also we can generate a library of labelled feature descriptors and then you can detection of classes using class labels. Here also you can use different classifiers, you can use matching to the nearest texture descriptors using a. So, matching to the nearest texture descriptor using some distance function.

(Refer Slide Time: 38:33)



So, with this let me stop here where, we have discussed different kinds of region descriptors. And we will continue this discussion still we will be discussing how globally also an image could be represented by a feature descriptor. So, let us stop at this point.

Thank you very much.

Keywords: Descriptors, matching, detection, feature