Computer Vision and Image Processing Fundamentals and Applications

Professor Doctor M. K. Bhuyan
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

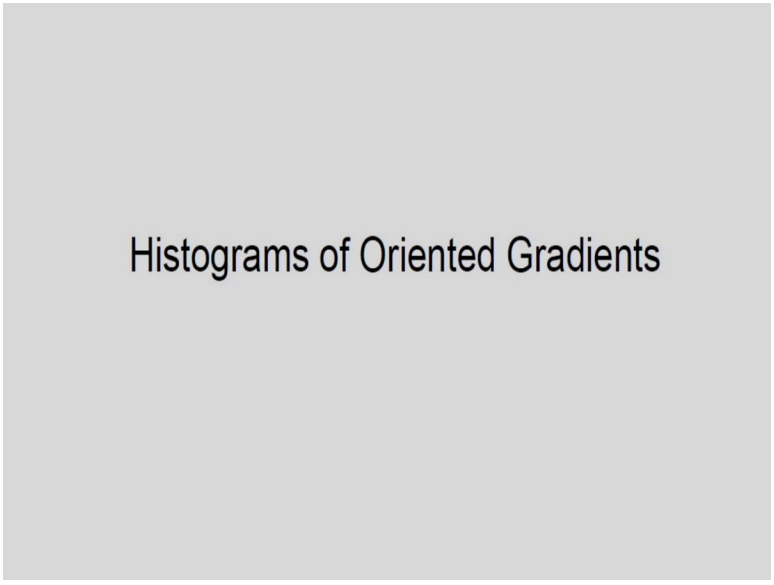
Lecture 29

Image Feature – HOG and SIFT

Welcome to NPTEL MOOCs course on Computer Vision and Image Processing Fundamentals and Applications. In my last class I discussed the concept of interest point detection, today I am going to discuss about two important image feature descriptors one is hog that is the histogram of oriented gradients and another one is sift the scale invariant feature transformation. In case of the hog we can determine gradient of an image, the orientations of gradients give information of the image that we can consider as image feature.

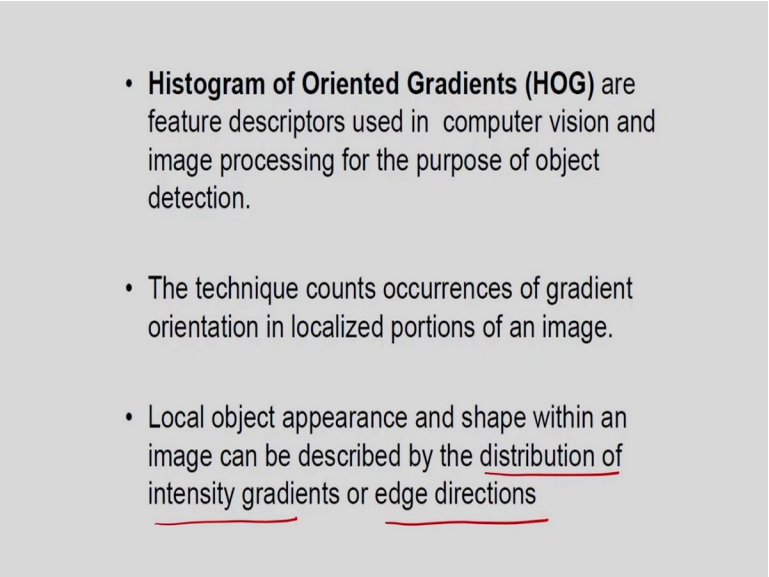
In case of the sift the objective is to determine image features or the extract image features which are invariant to scale and rotation, also the feature should be robust to viewpoints changes. So, these are the objective of sift features. So, these two concepts one is the hog another one is sift that I am going to discuss in this class. So, what is your hog? Let us, see.

(Refer Slide Time: 01:43)



So, histogram of the oriented gradients.

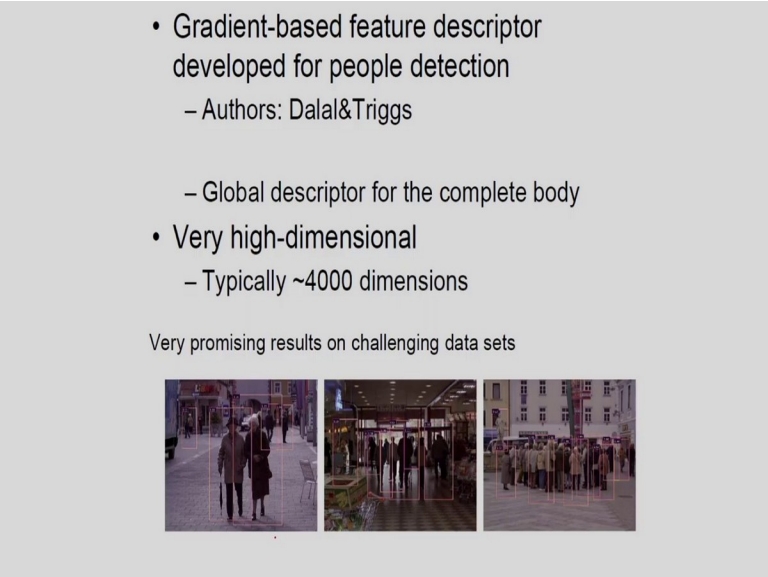
(Refer Slide Time: 01:46)



So, it is a features descriptors used in computer vision applications and one application is object detection. So, in this case we have to count the occurrences of gradient orientation in localized portions of an image and the local object appearance and set within an image can be described by the distribution of intensity gradients or edge directions.

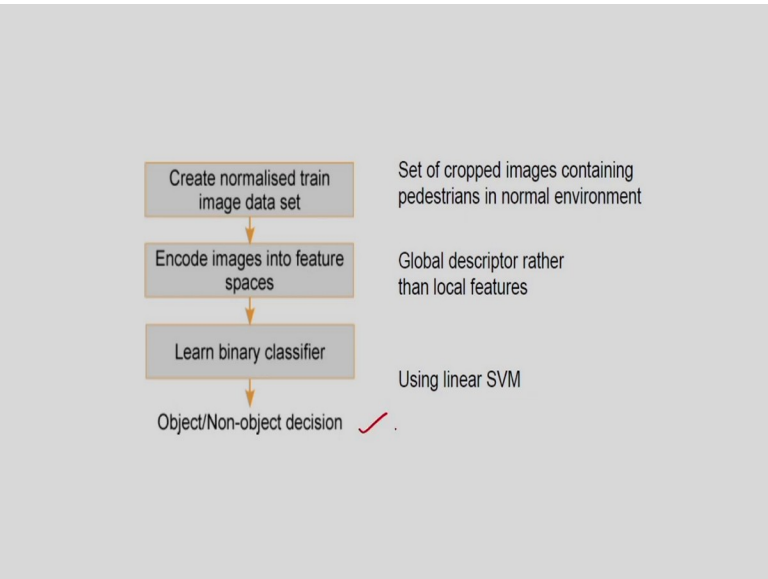
So, main concept is I have to determine the gradient orientations in localized portions of an image and also we have to count the occurrences of gradient orientation in localized portion of an image. So, that is the main concept of the histogram of the oriented gradient. So, that means the main concept is we want to see the distribution of intensity gradients or edge directions and by using this I want to represent a particular image.

(Refer Slide Time: 02:42)



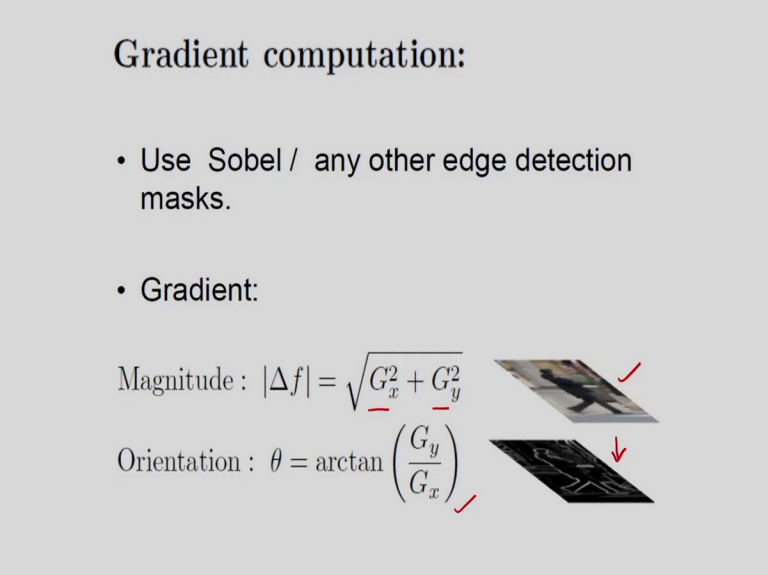
So, gradient based feature descriptors where develop for people detection in the paper by Dalal and Triggs they considered gradient-based feature descriptors for people detection. And this is the global descriptor for the complete human body and very high-dimensional typically 4000 dimensional. So, very promising results on challenging data sets. So, here you can see I am showing some of the results of people detection by considering gradient base feature descriptors. So, you can see the results like this.

(Refer Slide Time: 03:20)



So, for this what we have considered for this the procedure is like this, create normalised train image data set. So, first I have to create the normalised training image data set that is set up crop images containing pedestrians in normal environment, is the first step. So, training image data set we have to create, after this encode images into features spaces, so for this we are considering the global descriptor rather than local descriptors there local features, after this we have to learned a binary classifier, maybe we can consider the support vector machine and based on this we can detect objects or the non-objects that is mainly the object detection. So, this is a simple procedure.

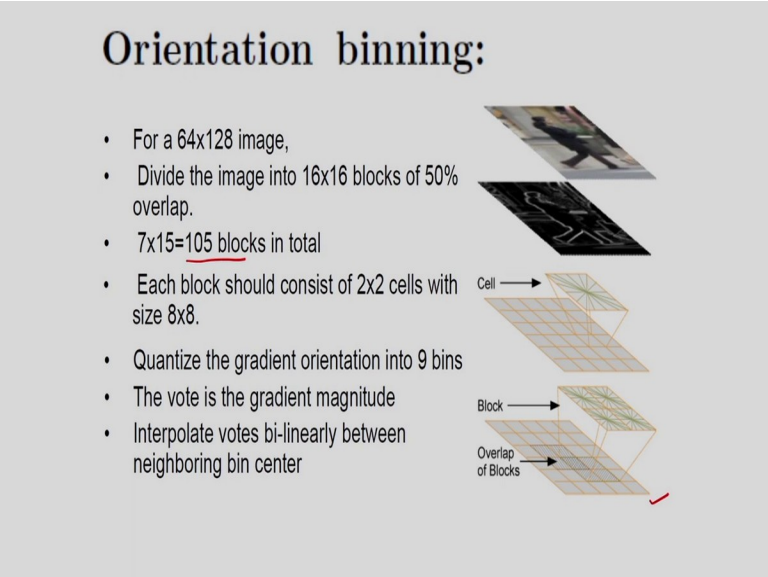
(Refer Slide Time: 04:08)



So, in case of the hog, the procedure is like this, we can consider the this mask maybe it is something like the Sobel mask we can consider or maybe any other edge detection mask we can consider and by considering these mask we can determine gradient magnitude. So, for this we have to determine gradient along the x direction and the gradient along the y direction we have to determine and based on this we can determine the gradient magnitude.

So, corresponding to this image you can see I have determined a gradient magnitude here. And also I can determine the orientation that is the direction of the normal to the edge I can determine so theta is nothing but Tan inverse Gy divided by Gx that is the direction of the A's normal that I can determine. So, this is the first step.

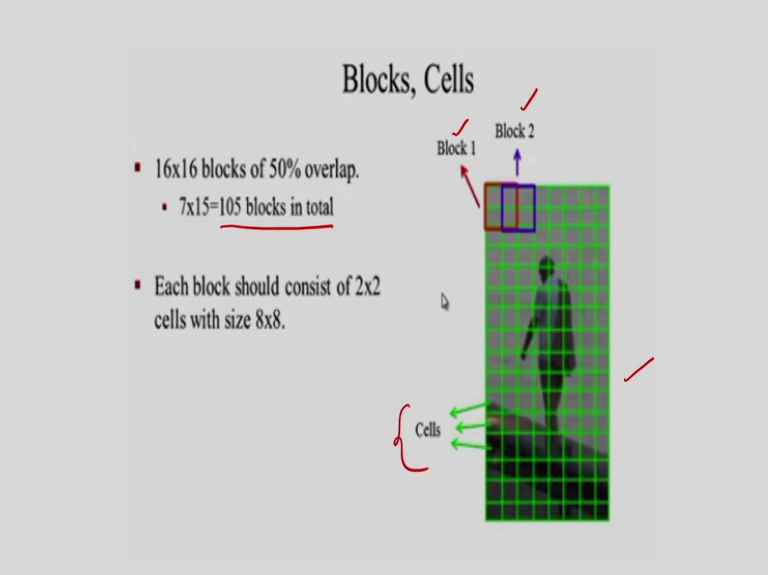
(Refer Slide Time: 04:59)



After this next what we have to do that is called the orientation binning, so for a 64 cross 128 image divided the image into 16 cross 16 blocks of 50 percent overlap, so you can see corresponding to the image I am considering 16 cross 16 blocks of 50 percent overlap, so based on this I will be getting 105 blocks total the total blocks will be 105 blocks and each block should consist of 2 cross 2 cells which size 8 cross 8, after this the quantize the gradient orientation into 9 bins, so we are considering 9 bins, that means 9 gradient orientations I am considering.

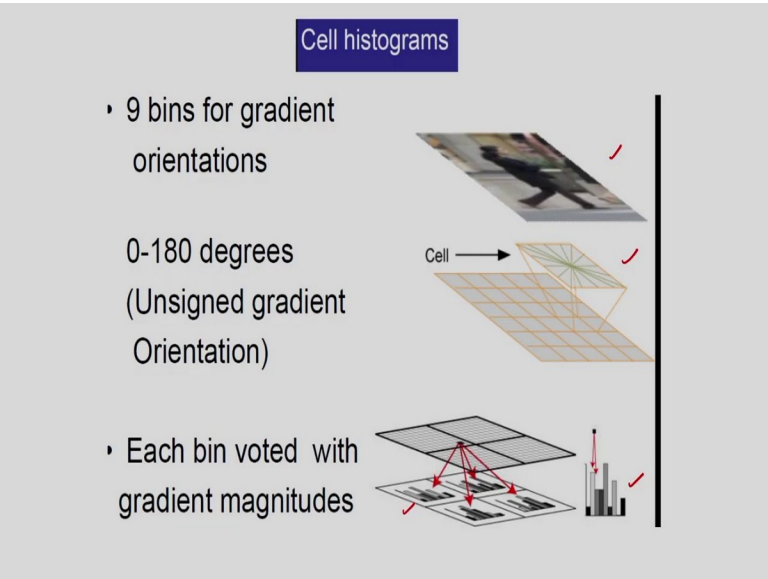
The vote is the gradient magnitude, so that we have to do that is a voting is the gradient magnitude and interpolate both by linearly between neighbouring bin centres. So, we have to do the interpolation that is a bilinear interpolation between neighbouring bin centres. So, this is the concept of the orientation binning, so that concept again I am going to explain in my next slide.

(Refer Slide Time: 06:11)



So, here you can see I am showing the blocks and the cells corresponding to this input image and I am considering the 50 percent overlap so you can see the block 1 and the 2 that is 50 percent overlap, so in this image I will be getting 105 blocks in total. And you can see I am considering the cells, so each block should consist of 2 cross 2 cells, so you can see the cells here with size 8 cross 8.

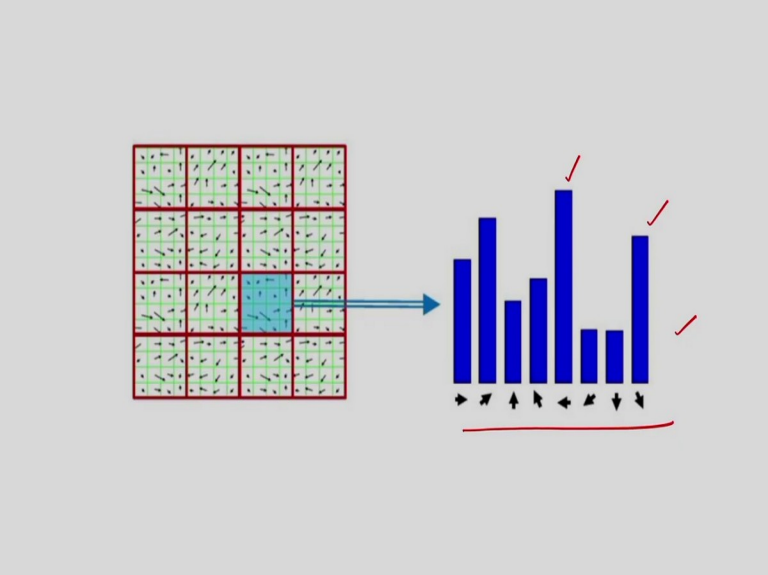
(Refer Slide Time: 06:44)



So, I have 9 gradient orientation directions that means the 8 bins for gradient orientations and 0 to 180 degree we are considering so unsigned gradient orientation from 0 to 180 degrees, each

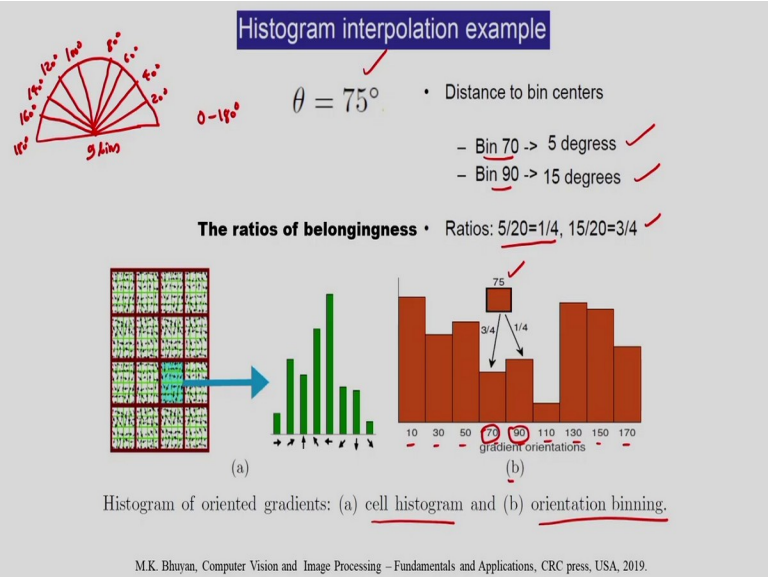
bin voted with gradient magnitude. So, in the figure you can see I am showing the cells corresponding to this input image and I have the histograms of oriented gradients and you can see this is the histogram of oriented gradients. So, each bin voted with gradient magnitudes.

(Refer Slide Time: 07:19)



So, here corresponding to this image and I am showing the blocks you can see the histogram, the histogram of the oriented gradients. So, these are the orientations I am considering, so for different orientations I have to go for voting. So, corresponding to this orientation this is the value after voting corresponding to this orientation so I am just doing the voting and that is the magnitude of that particular histogram and like this corresponding to this orientation this is the magnitude. So, the vote is the gradient magnitude.

(Refer Slide Time: 07:53)



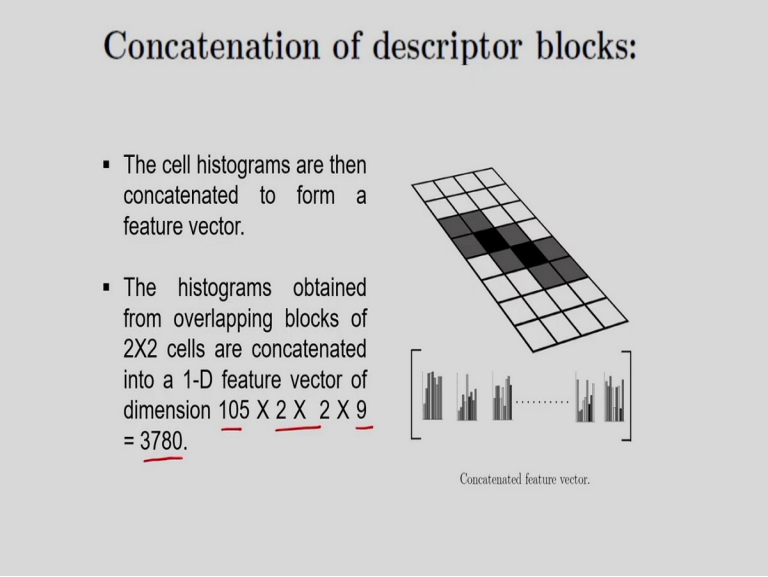
So, that means if I consider 0 to 180 degree, that means I am considering the gradient orientations and I am considering 9 bins, so suppose I am considering 0 to 180 degree, so this 0 to 180 degree I am considering, so suppose this is 20 degree, this is 40 degree, this is 60 degree, this is 80 degree, this is a 100 degree, this 120 degree, 140 degree, 160 degree and it is 180 degree. So, that means we are considering 9 bins, we are considering 9 bins.

So, each block has 2 cross 2 cells which size 8 cross 8 and quantize the gradient orientation into 9 bins, so we are considering from 0 to 180 degree. The vote is the gradient magnitude and also we have to interpolate vote's sides by linearly between the neighbouring bin centres. So, corresponding to theta is equal to 75 degree distance to bin centres, so if I consider bin 70 the distance to bin centre will be 5 degrees and if I consider bin centre 90, so distance to bin centre will be 15 degree and after this we have to determine the ratio of the belongingness, so the ratio will be 5 divided by 20 that is nothing but 1 divided by 4.

And another one is 15 divided by 20 that is nothing but 3 divided by 4. So, that means I am considering the ratio of the belongingness that is what I am considering interpolating votes by linearly between the neighbouring bin centres. So, corresponding to 75 the neighbouring bin centres are 70 and the 90. So, based on this I am determining the ratio of belongingness. So, in the figure a you can see I am showing the cell histograms.

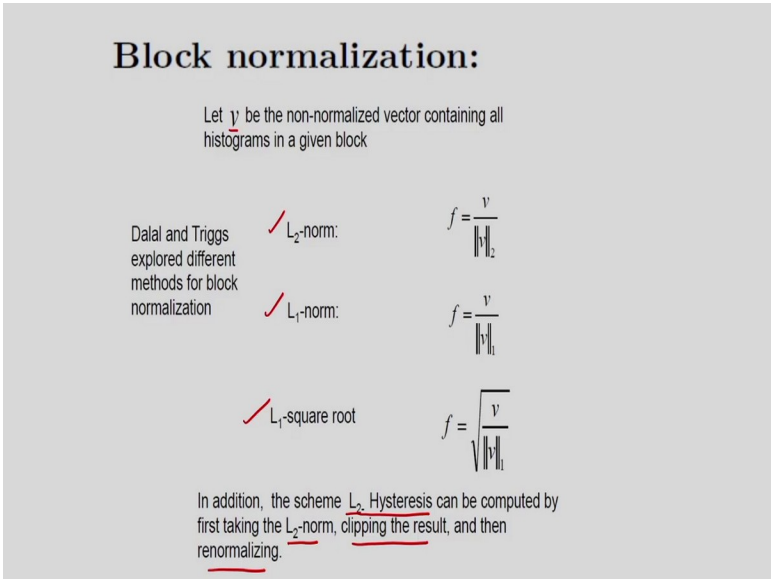
And in the figure b what I am showing? I am showing the orientation binning. So, I am considering 9 bins. So, you can see in the histogram you can see the 9 bins, 1, 2, 3, 4, 5, 6, 7, 8, 9, so 9 bins I am considering. So, this is about the orientation binning and how to do the interpolation the histogram interpolation.

(Refer Slide Time: 10:38)



After this we have to determine the feature vector, so for this what we are doing? That concatenation of descriptor blocks, the cell histograms are then concatenated to form a feature vector. So, histogram obtained from the overlapping blocks of 2 by 2 cells are concatenated into 1-D vector of dimension you can see we have 105 blocks and the 2 cross 2 cells 9 bins, so that means the dimension of the feature vector will be 3780.

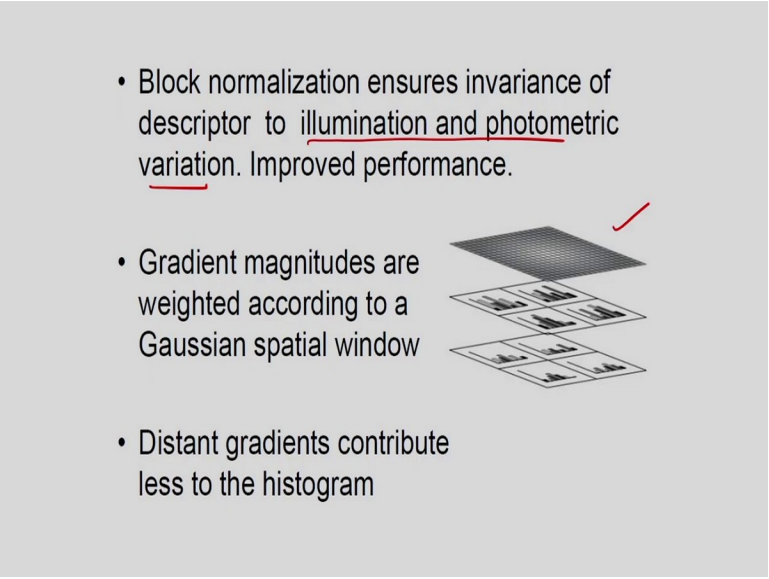
(Refer Slide Time: 11:13)



After this we have to do the block normalization, so the in the paper by Dalal and Triggs they explode different methods for block normalization, so maybe we can consider L2 norm L1 norm or the L1 square norm we can consider I am considering v is a non-normalized vector containing all histograms in a given block. After this I can do the normalization that I can consider L2 norm L1 norm or maybe L1 square root I can consider.

In addition the scheme L2 hysteresis can also be computed, how to compute this? The compute means the L2 hysteresis I can compute first by taking L2 norm clipping the result and after this we have to do the renormalization. So, by this procedure we can determine the L2 hysteresis. So, this is about the block normalization.

(Refer Slide Time: 12:14)

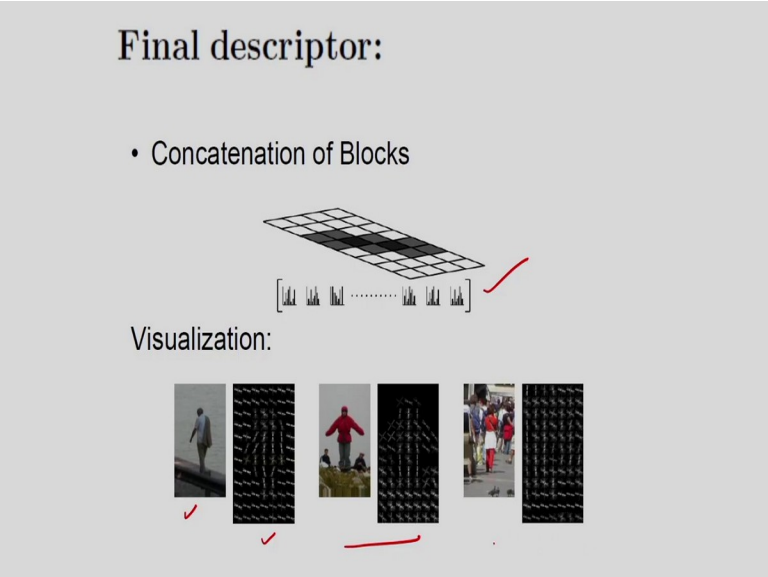


What is the importance of the block normalization? The block normalization ensures invariance of descriptor to illumination and the photometric variations. So, it improved the performance, so for this Illumination in the photometric variation we have to consider the block normalization. The descriptors should be invariant to illumination and the photometric variations. So, that is why we have to do the block normalization.

After this, the gradient magnitude are weighted according to a Gaussian special window, so we can consider a Gaussian special window we can consider and the gradient magnitude can be weighted based on this Gaussian special window. Distant gradient contribute less to the histogram, so that concept we are considering because if I consider the boundary, so we have to neglect the boundary, the gradient corresponding to the boundary.

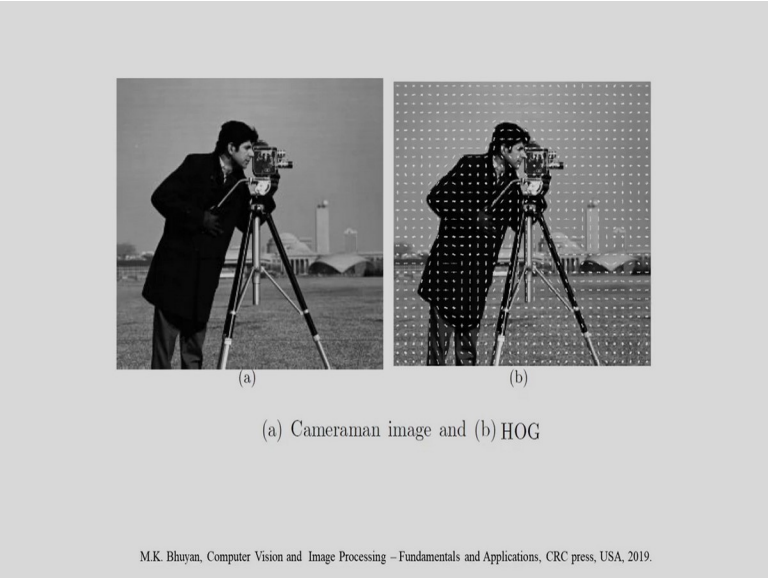
So, that is why the distant gradient contribute less to the histogram. So, that is the objective of the weighted gradients. So, in the figure I am showing the Gaussian special window, so this is the Gaussian special window I am considering and the gradient magnitude are weighted by this Gaussian special window.

(Refer Slide Time: 13:32)



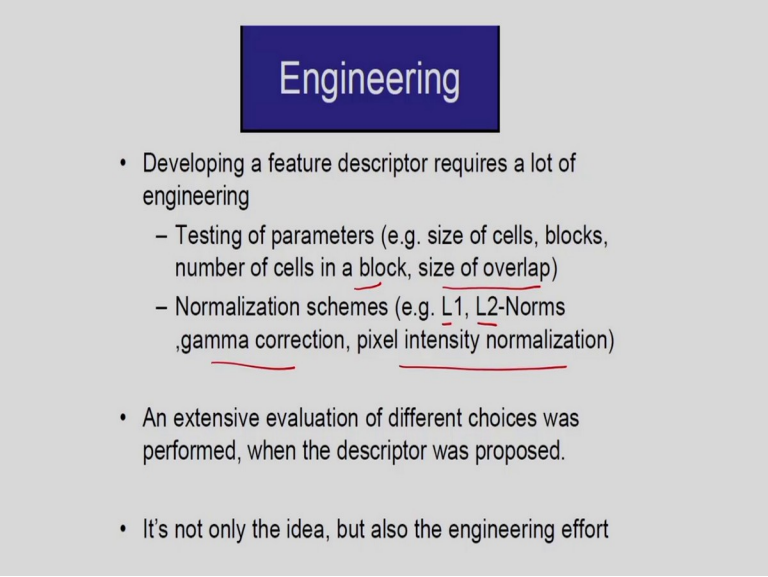
So, finally I am getting the hog feature that is the concatenation of the blocks, so this is the feature vector I am getting. And for visualization you can see corresponding to this input image this is the histogram of the oriented gradient, corresponding to the second image you can see the histogram of the oriented gradient and like this you can see and corresponding to the third image also we can see the histogram of the oriented gradients.

(Refer Slide Time: 13:58)



And similarly in this case if you see the input image with the cameraman image and the second image is the hog the histogram of oriented gradients. So, by using the hog, I can represent the input image.

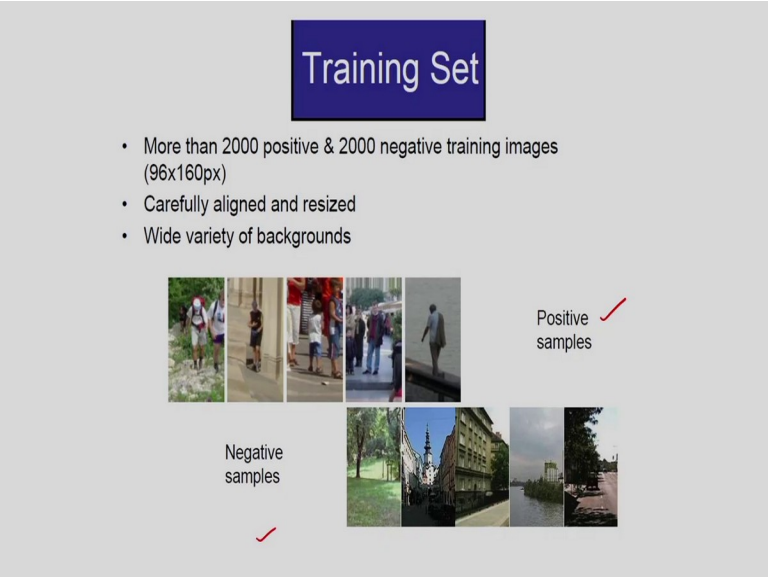
(Refer Slide Time: 14:11)



For hog there where many engineering efforts, so this is called a feature engineering, so like this the testing of the parameters the parameters may be the size of the cells, the blocks number of cells in the a block and the size of the overlapping, so these are the parameters. Also the normalization schemes we can consider L1 norm L2 norms like this or maybe we can consider gamma correction pixel intensity normalization, so we have to do the testing for all these parameters.

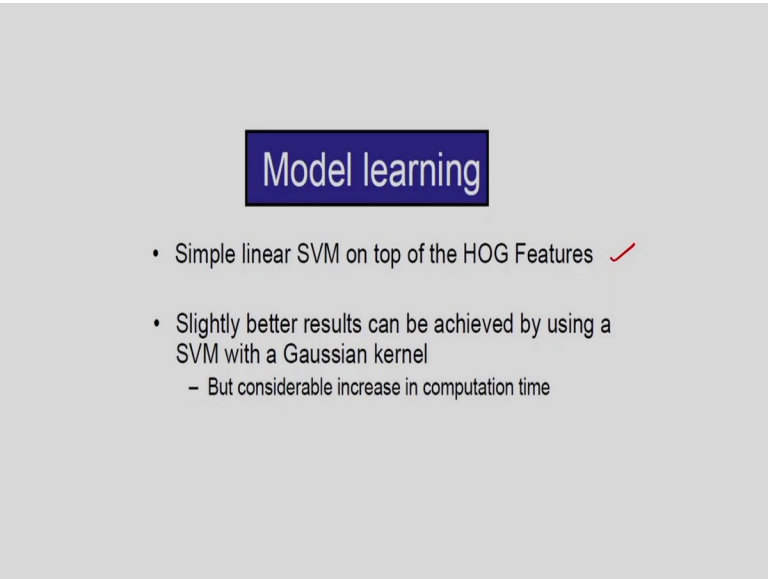
And extensive evaluation of different choices was perform when the descriptor was proposed. So, that means many test where done to see the performance of the hog descriptors and based on this these parameters were fixed, the parameters may be the size of the cells, the block size, number of cells in a block like this. So, these parameters will fix after all the experimentations. So, it is not only the idea, but also the engineering effort.

(Refer Slide Time: 15:12)



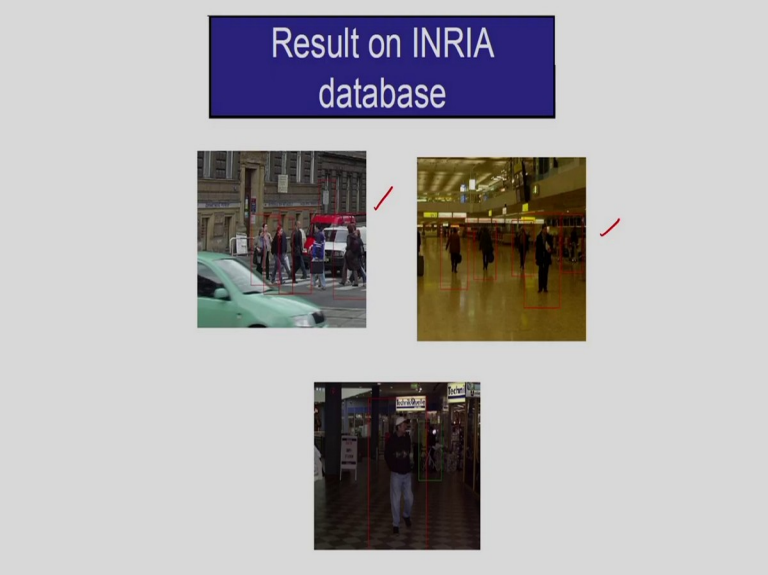
And the training set was like this, so more than 2000 positive images and 2000 negative training images were reconsidered and all these images are aligned and resize and they considered wide variety of backgrounds. So, you can see there are some positive training images and the negative training images and also considered different types of backgrounds, so different backgrounds they considered.

(Refer Slide Time: 15:41)



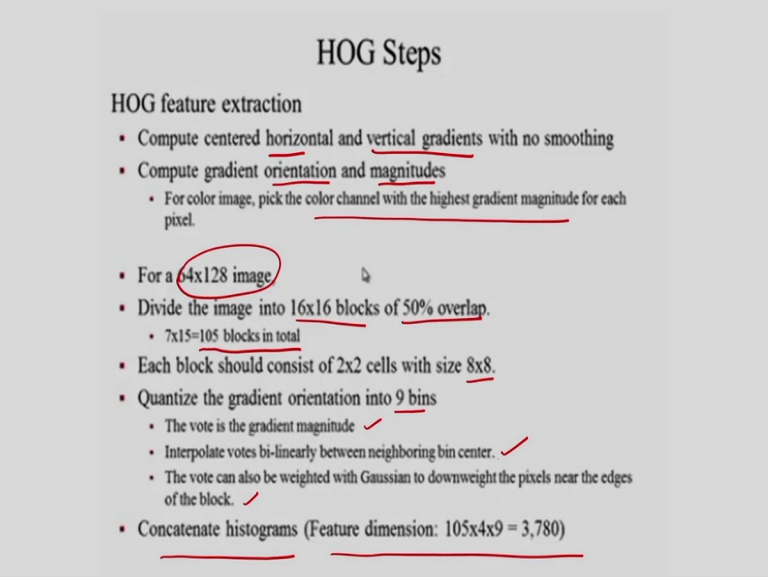
After this the support vector machine was used on the top of the hog features and slightly better results can be achieved by considering the support vector machine with a Gaussian kernel. But it increases the computational complexity. So, this is about the model learning.

(Refer Slide Time: 15:58)



Some of the results of the INRIA database you can see these are the output images that means the people detection, so these are the results of the people detections corresponding to these images you can see.

(Refer Slide Time: 16:12)

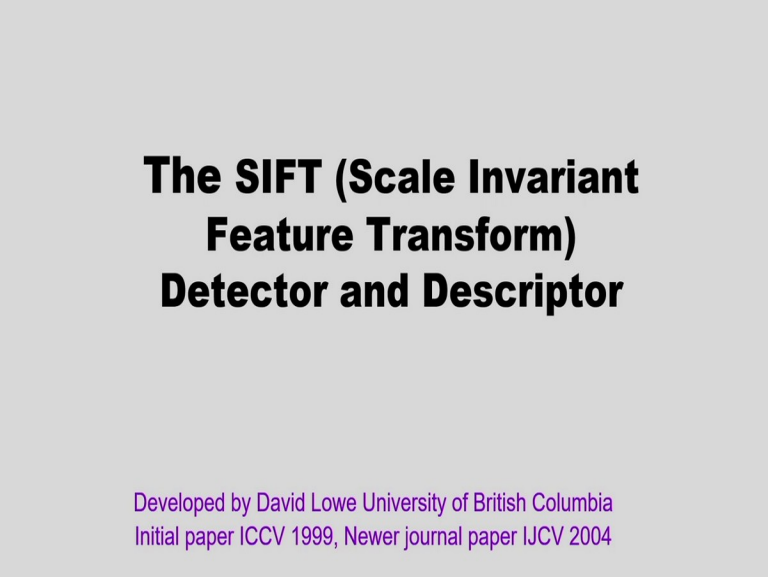


So, what are the important steps of the hog descriptors? I am again explaining so the procedure of hog feature extraction is first compute the centered horizontal and the vertical gradient with no smoothing first I have to compute the horizontal and vertical gradients, after this we have to determine the gradient magnitude and the orientations. So, if I consider a colour image, what is the procedure? Pick the colours channel with the highest gradient magnitude for each pixel, so for colour image we have to consider this one with the highest gradient magnitude for each pixel.

So, for 64 cross 128 image divide the image into 16 cross 16 blocks of 50 percent overlap that we are considering, so if I consider this one, so there will be 105 blocks in total and each block should consist of 2 cross 2 cells with size 8 cross 8, after this quantized the gradient orientation into 9 bins that already I have explained, the vote is the gradient magnitude and I have to do the interpolation the interpolate votes by linearly between the neighbouring bin centres, so that concept also I have explained.

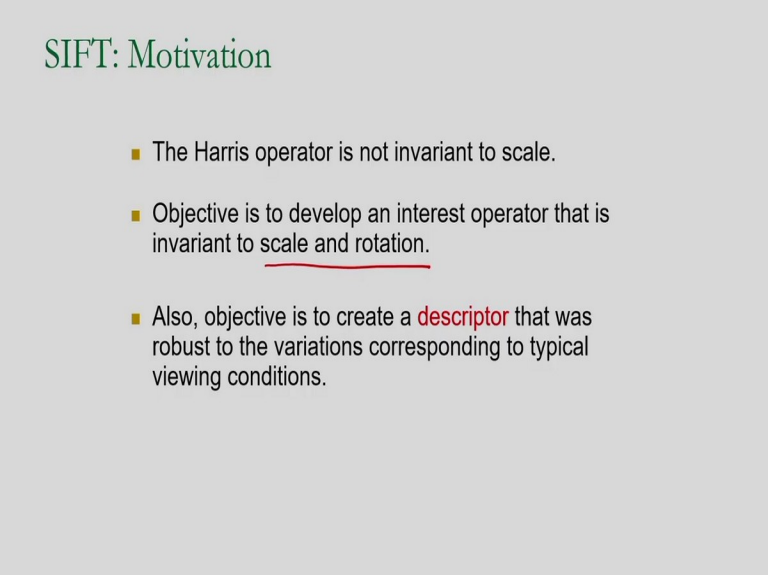
And after this the vote can also be weighted with Gaussian to down weight the pixels near the edges or the boundaries of the block that we are considering and after this concatenating histograms the dimension of the feature will be 3780, so that is the dimension of the feature corresponding to the image, the image is 64 cross 128 image. So, I will be getting the hog features. So, this is the fundamental concept of the histogram of the oriented gradient.

(Refer Slide Time: 18:05)



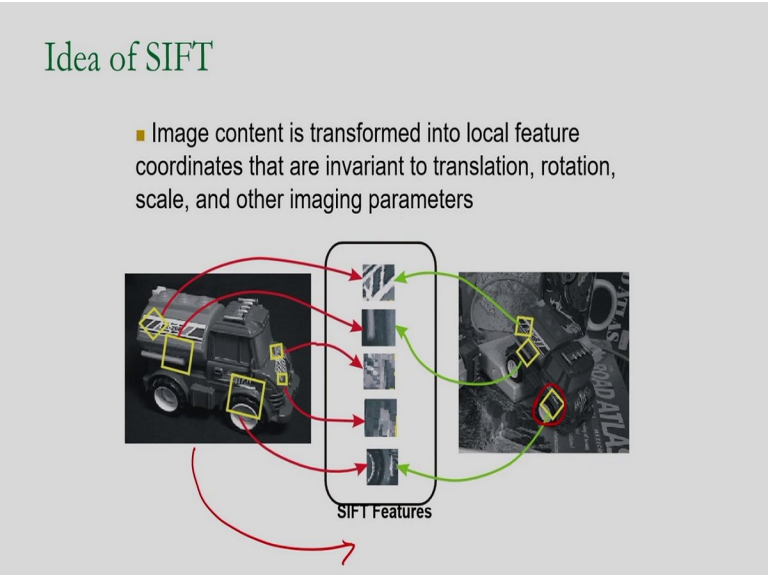
Now, I will discussed about the SIFT that is the scale invariant feature transformation, so this concept was developed by David Lowe of university of British Columbia.

(Refer Slide Time: 18:16)



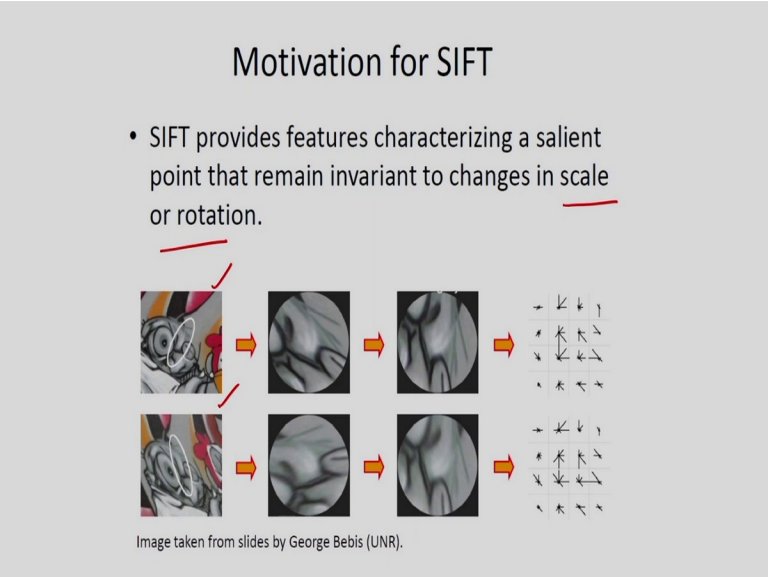
So, you know that Harris operator is not invariant to scale, so the objective is to develop an interest operator that is invariant to scale and rotation, so that is the one objective of sift the scale invariant feature transformation. So, the feature should be invariant to scale and rotation. Also the objective is to create a descriptor which is robust to the variation corresponding to typical viewing conditions. So, that is another objective of sift.

(Refer Slide Time: 18:48)



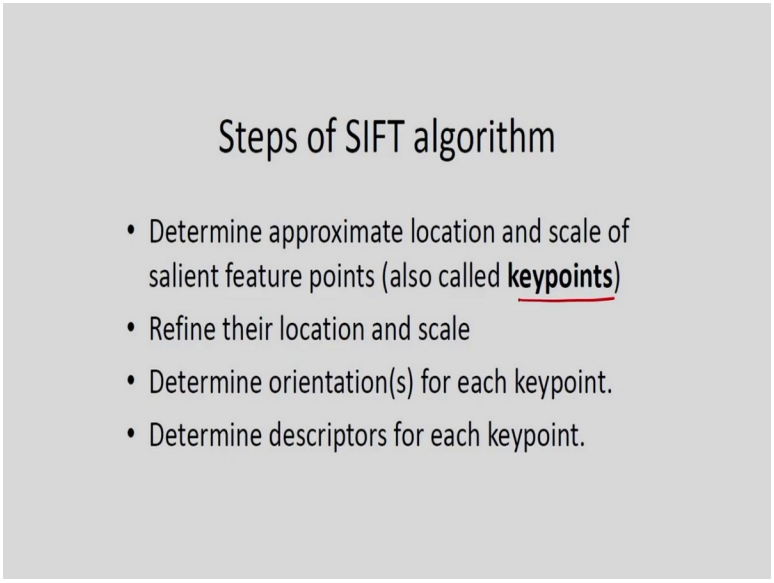
So, if you see here I am showing one image and I am considering the image features which are invariant to translation rotation scale and other imaging parameters. So, from this input image I am extracting the features these are the 6 features and these features should be invariant to translation rotation scale and other imaging parameters. So corresponding to this image you can see I am considering some features or the key points that key points are invariant to translation rotation scale and other imaging parameters. So, that I can consider as sift features or the descriptors.

(Refer Slide Time: 19:29)



So, SIFT provide features characterizing a salient point that remain invariant to changes in scale or rotation that is the main objective of the SIFT that means the feature should be invariant to changes in scale or rotation. So, here you see corresponding to this input images I am extracting the SIFT feature which are invariant to changes in scale or rotation. So, these I can consider as SIFT feature.

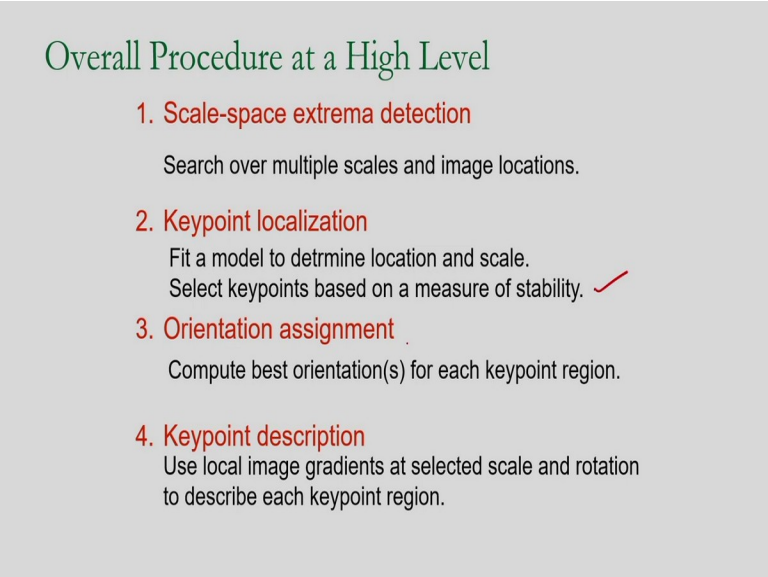
(Refer Slide Time: 20:01)



Now, I will highlight the steps of SIFT algorithm. The first one is the determine approximate location and scale of the salient feature points. This salient feature points are called the key points, after this we have to refine the location and the scale of the key points. So, already we have extract the key points and after this we have to refine the location and the scale of the key points.

After this, we have to determine orientations for each of the key points, because already we have extract the key points and we have to determine the orientations for each of the key points that is the third step. After this finally we have to determine descriptor for each of the key points. So, these are the main steps of the SIFT algorithm.

(Refer Slide Time: 20:50)

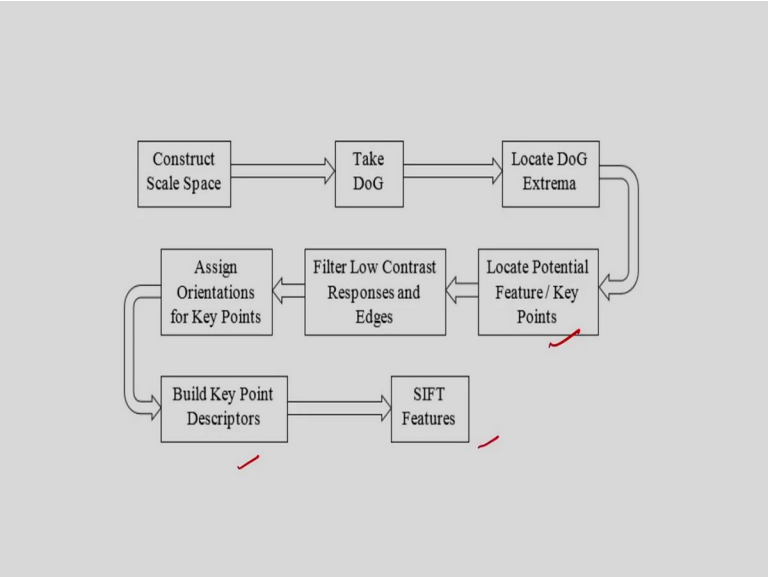


So, overall procedure at a high level first I have to consider scale-space extrema detection that means I have to search over multiple scales and different image locations that means I have to find the key points. So, for this we are considering that is a searching over multiple scales and different image locations. The next step is the key point localization that means fit a model to determine location and scale of the key points and based on this select key points based on a measure of stability, so based on this we can select the key points.

After this orientation assignment that is the third step, so compute base orientation for each key point region, because already we have extracted the key points and also after extracting the key points we have to compute best orientation for each key point regions. And finally in the step 4 key point description, so for this we can use local image gradient at selective scale and rotation to describe each key point region.

So, these are the steps of the SIFT algorithm, I am repeating it again. So, first I have to consider the searching over multiple scale and different image locations, after this we have to find the key points and after this orientation assignment means computer the best orientation for each of the key point regions and finally the description for the key points.

(Refer Slide Time: 22:36)

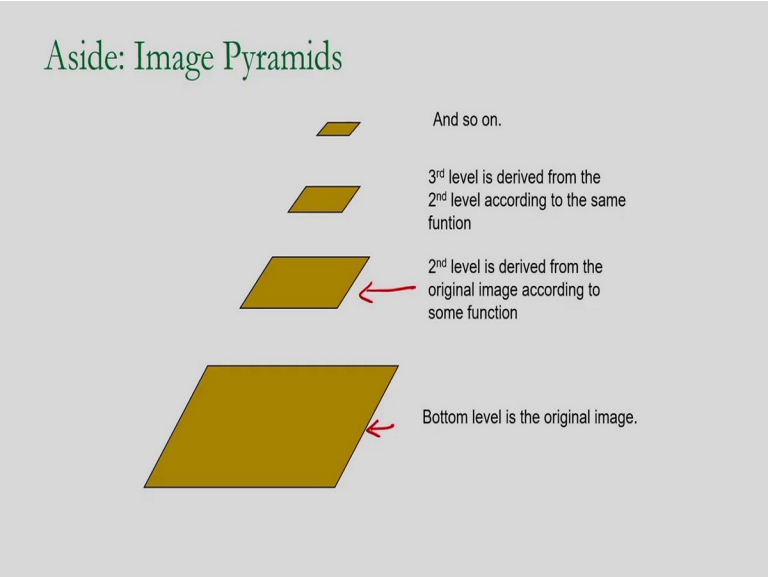


So, in the block diagram I have shown the algorithm that a SIFT algorithm I have shown, so first I have to construct the scale space and after this we have to consider the difference of Gaussian I am considering and locate the DoG extrema that means in case of the difference of Gaussian I can determine the extreme points, this is the maximum point I can determine or the minimum point I can determine in the DoG image that is a difference of Gaussian image, so I can find a maximum or the minimum and based on this I can find the key points.

So, you can see I can find the or I can locate at the potential feature potential key points. After this what we have to do? We have to filter out the low contrast responses that may not be the key points, so that is why we have to filter out low contrast responses and also we have to neglect the edge pixels which are considered or which are detected as key points, because edges or the edge pixels are not the key points some of the edge pixels may be detected as key points.

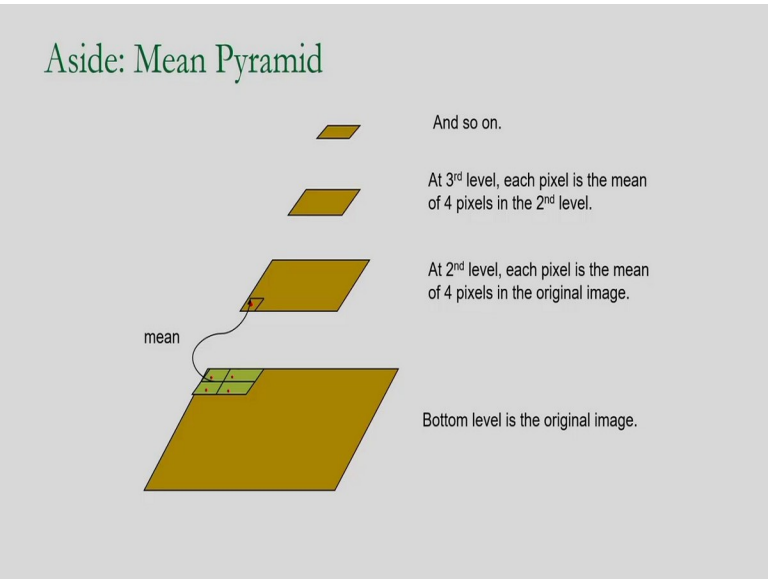
So, that we have to eliminate. So, that is why the filtering out low contrast responses and the edge pixels. After this assigning orientations for key points and after this we have to find the descriptors for the key points and that is nothing but the SIFT features. So, this is the block diagram of the SIFT algorithm.

(Refer Slide Time: 24:11)



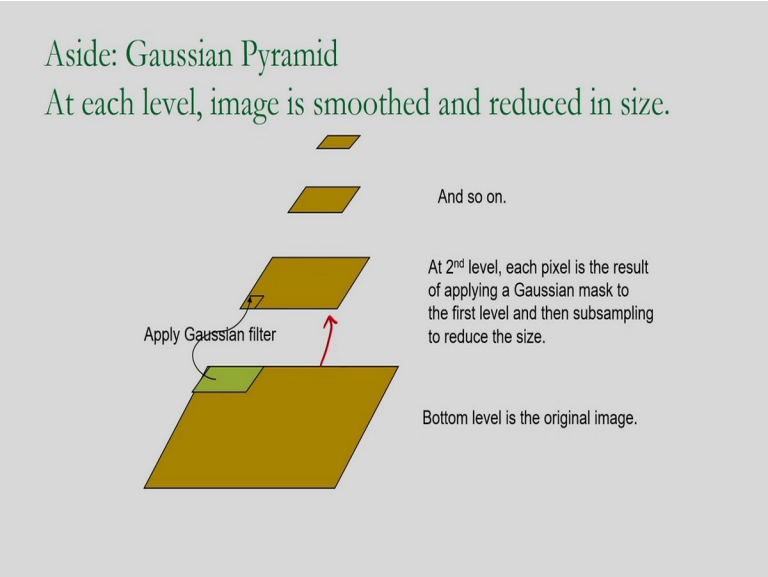
So, first I will discuss about image pyramids. So, here I have shown one image pyramid, so here you can see the bottom level is the original image, so the first one is the original image, second level is derived from the original image according to some functions, so I am getting the second level that is obtained from the first image, the original image. And third level is obtained from the second level according to some functions and like this I will be getting all these levels and that corresponds to the image pyramid.

(Refer Slide Time: 24:49)



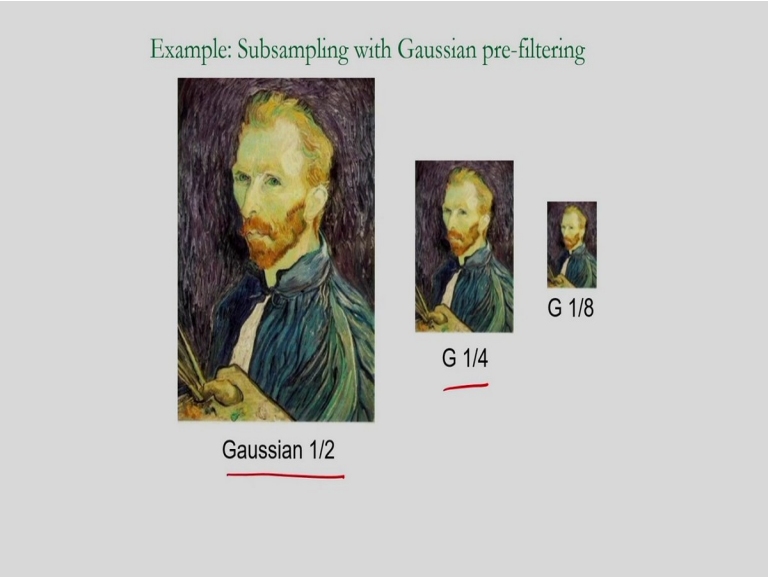
Now, let us discuss about mean pyramid. So, here you can see the bottom level is the original image, at second level each pixel is the mean of 4 pixels in the original image. So, that means I am considering 4 pixels like this and this pixel is nothing but the mean of 4 pixels of the original image. Similarly, at third level each pixel is the mean of 4 pixels in the second level and like this we have to go on and we will be getting the mean pyramid like this. So, this is one example of the image pyramid. In the next slide also I am showing another pyramid that is Gaussian pyramid.

(Refer Slide Time: 25:31)



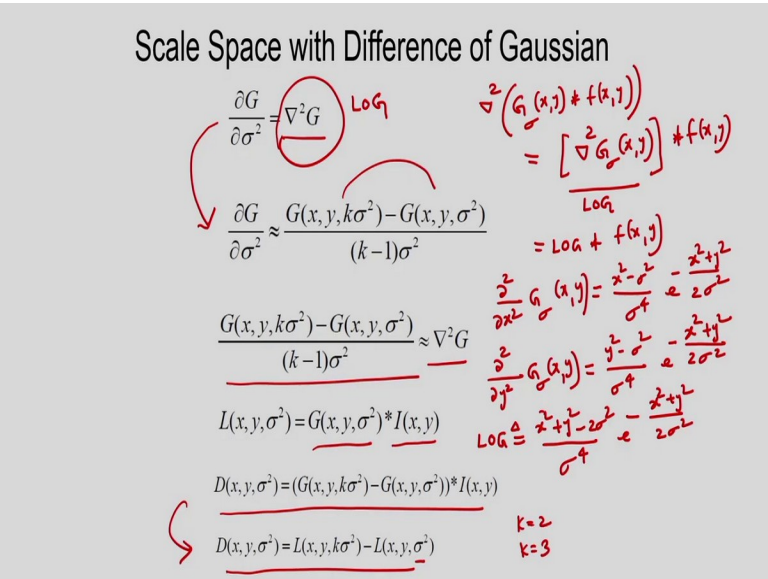
So, that is the Gaussian pyramid. So, bottom level is the original image, after this we can apply a Gaussian filter, so if I apply Gaussian filter the image will be smooth, the image will be blurred and after this at a second level each pixel is the result of applying a Gaussian mask to the first level and then resampling to reduce the size. So, that means this image is resampled and I will be getting the second level and like this I will be getting all the levels of the Gaussian pyramid.

(Refer Slide Time: 26:08)



In this example I am showing the subsampling with Gaussian pre-filtering, so this is Gaussian I am showing it is a half, the Gaussian image is down sample by a factor of 2 to produce an image that is the image is this 1 4th I am getting 1 4th size I am getting like this I have to do the down sampling. So, the image is convolved with a Gaussian and after this I am doing the down sampling. So, this is the concept of the subsampling with a Gaussian pre-filtering.

(Refer Slide Time: 26:39)



With the difference of Gaussian we can construct the scale space. So, in one of my class I discussed about the Laplacian of Gaussian, so here you see this is nothing but the Laplacian of

Gaussian that we can determine, so what is the Laplacian of Gaussian? That is the this is the Laplacian and suppose I am considering a Gaussian function G x y that is convolved with the image the image is f x y and that is equal to so this is nothing but the Laplacian of Gaussian, this is the log the Laplacian of Gaussian.

So, I can write like this, this is the Laplacian of Gaussian and this is convolve with f x y, f x y is the image. So, for calculating the Laplacian of Gaussian, we can determine the differentiation with respect to x that is a second order differentiation of the Gaussian function, so this is equal to x square minus sigma square sigma to the power 4 e to the power minus x square plus y square 2 sigma square. So, I can determine the second order derivative with respect to x for the function the function is the Gaussian function.

Similarly, we can also determine the second order derivative with respect to y corresponding to the Gaussian function the Gaussian function is G sigma x y, so that will be also equal to y square minus sigma square sigma to the power 4 e to the power minus x square plus y square 2 sigma square. And finally from these 2 second order derivative I can determine the log.

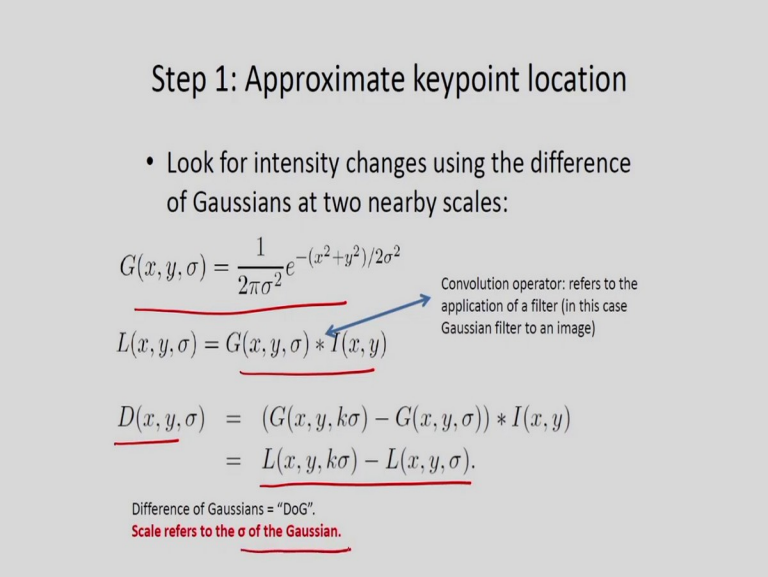
So, log will be like this that is the Laplacian of Gaussian, so it is approximately equal to x square plus y square minus 2 sigma square sigma to the power 4 e to the power minus x square plus y square 2 sigma square, so I can determine the Laplacian of Gaussian. So, in this case you can see this is nothing but this is the Laplacian of Gaussian. So, this Laplacian of Gaussian here you can see just I am determining this.

So, this is nothing but the difference between two Gaussians, in this case that this parameter k controls the scale, because the sigma actually the sigma means the scale, sigma of the Gaussian function it corresponds to the scale. So, in one function the sigma square is multiplied by k and in another function it is only sigma square. So, based on that k I can change the scale.

So, corresponding to this Laplacian of Gaussian, so I will be getting this expression and here you can see the Gaussian function is convolve with the image and in this case if I do the convolution the image will be smooth and corresponding to this I can determine the difference of Gaussian, so this is the difference of Gaussian that is nothing but I will be getting this, l xy k sigma square minus l x sigma square. So, sigma controls the scale.

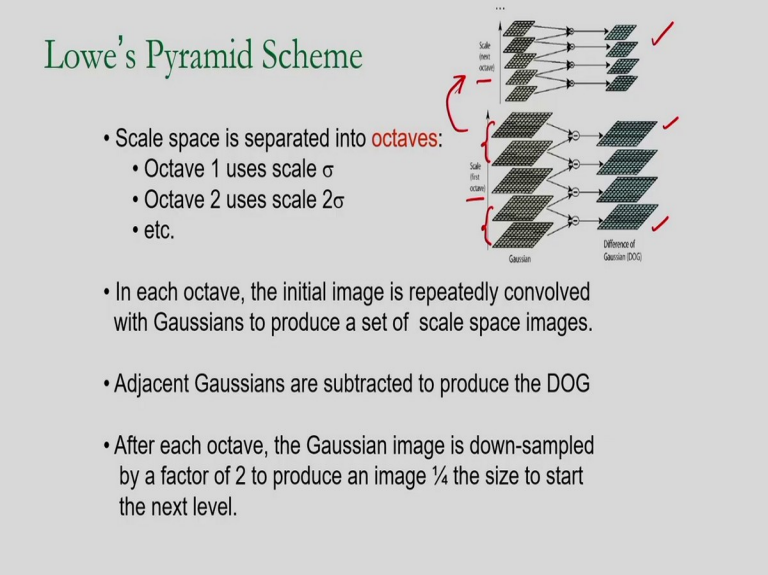
So, this is the concept of the difference of Gaussian. So, with the difference of Gaussian, I can construct the scale space, because here you can see in this case the sigma square only the sigma square, but in this case xy k sigma square, so suppose if I put k is equal to 2, so I will be getting different scale, if I put k equal to 3, so I will be getting the different scale like this, I will be getting a number of scales. So, I can determine the difference of Gaussian.

(Refer Slide Time: 31:11)



So, first tip of the sift algorithm is approximate key point location. So, look for intensity changes using the difference of Gaussian at two nearby scale. So, this is the first step. So, this is the Gaussian function I am showing the two-dimensional Gaussian function and after this the image is one vote with the Gaussian that means I x y is the image and image is convolved with a Gaussian and because of this the image will be smooth and after this you can determine the difference of Gaussian. So, difference of Gaussian will be something like this and the scale corresponds to sigma of the Gaussian.

(Refer Slide Time: 31:57)



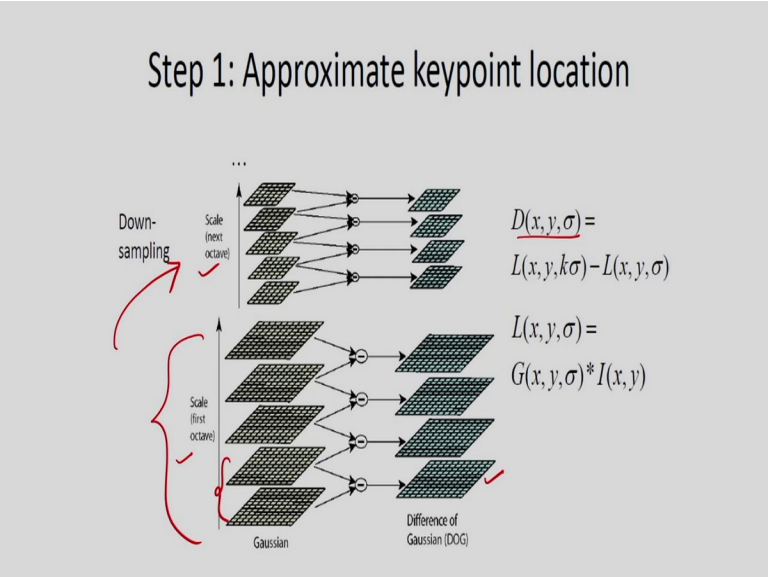
Here I am showing the pyramid scheme corresponding to the sift algorithm. Scale space is separated into octaves, so I have two octaves, so here you can see this is the first octave and this is the second octave like this the first octave uses scale sigma and the second octave uses the scale 2 sigma. So, in each octave the initial image that is the image the input image is repeatedly convolved with the Gaussian to produce a set of scale space images.

So, that means corresponding to this first octave what I am doing I am doing the convolution of the image with the Gaussian to produce a set of scale space images. After this the adjacent Gaussians are subtracted to produce DOG, so here you can see I am considering this to adjacent Gaussians and from this I can determine the difference of Gaussian like this.

Similarly, if I consider these two suppose these two and corresponding to this I can determine the difference of Gaussian. So, after this what I can do? The Gaussian image is down sample by a factor of 2 to produce an image 1 4th put the size to start the next level. So, that means I will be getting the second octave I will be getting or the next octave I will be getting for this I have to do the down sampling of the original image.

So, that means I am doing the down sampling and I will be getting the second octave, the next octave I will be getting. And corresponding to the next octave also I have to determine the difference of Gaussian.

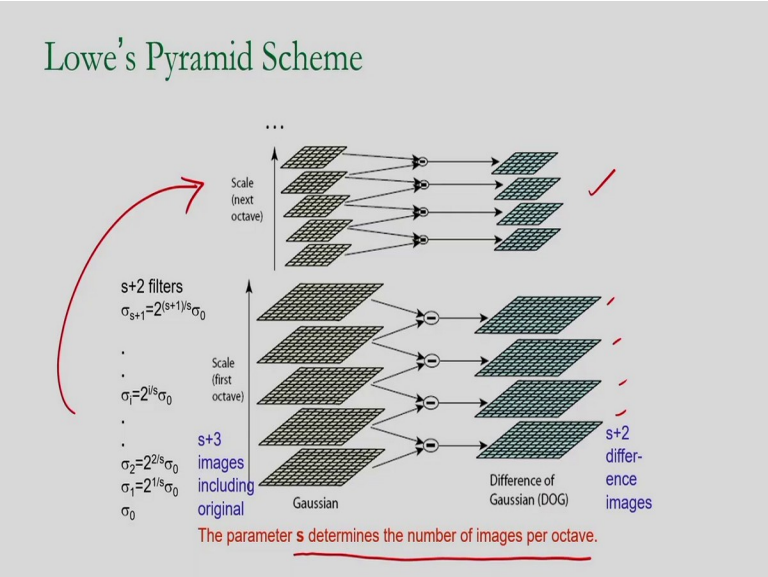
(Refer Slide Time: 33:40)



So, same thing here also I am showing here you can see I am showing two octaves that is the first octave I am showing and after this it is down sample, so I will be getting the next octave and here you can see I am finding the difference of Gaussian D x y sigma that is the difference of Gaussian I will be getting, so that means the image is repeatedly convolve with a Gaussian to produce a set of scale space images.

So, I will be getting these images, all these images I will be getting that means a set of scale space images I will be getting. After this what I can do? Adjacent Gaussians are subtracted so suppose these are subtracted to produce the difference of Gaussian, so I will be getting the differential of Gaussian like. After this the Gaussian image is down sample by a factor of 2, to produce an image of 1 4th of the original image. So, after this I am doing the down sampling and I will be getting the next octave. So, that is the concept of the scale space.

(Refer Slide Time: 34:51)



So, again I am showing the pyramid so you can see I am showing 2 octaves the first octave and the second octave and how to compute the difference of Gaussian also you can see here, so here you can see in the first octave I have s plus 3 images including the original, so suppose s is equal to 2 that means I will be getting 5 images in the scale 1 that is the in the first octave I will be getting 5 images corresponding to s is equal to 2.

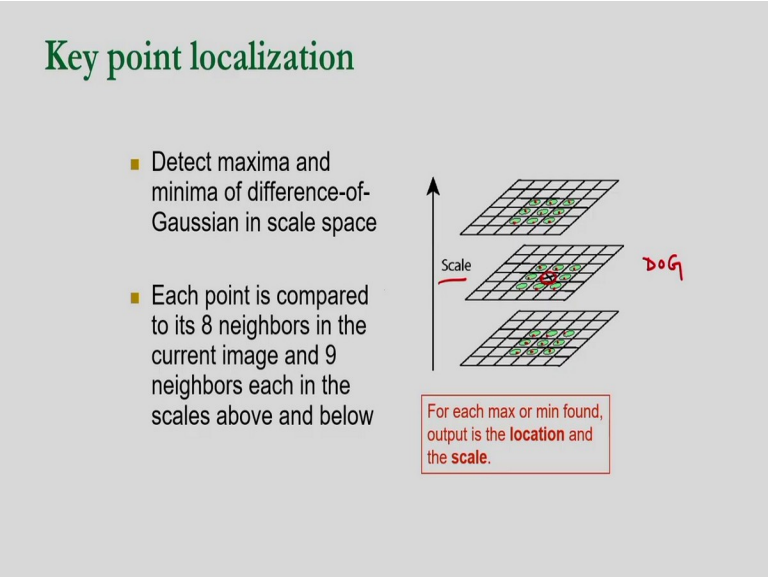
And how many different sub Gaussian images I will be getting? s plus 2 that means 2 plus 2 there will be 4, so 4 images I will be getting that is the difference of Gaussian. So, this parameter s determines the number of images per octave. And here you can see I am getting the number of images, how to get the number of images? That is the original image has repeatedly convolve with the Gaussians to produce a set of scale space images.

So, that concept already I have explained. So, corresponding to this parameter s you can see the number of images in the octave, so in the first octave suppose s is equal to 2, then I will be getting 5 scale space images and corresponding to this will be getting 4 difference of Gaussian images. And after this I have to do the down sampling after doing the down sampling I will be getting the second octave. And similarly I have to determine the difference of Gaussian. So, this is a pyramid scheme.

(Refer Slide Time: 36:31)

So, here I am showing the images corresponding to scale is equal to 0, scale is equal to 1, scale is equal to 4, scale is equal to 16, 64, scale 256. So, that means I am observing an image at different scales, that is the concept of the scale space representation and from this I want to determine the key points.

(Refer Slide Time: 36:54)



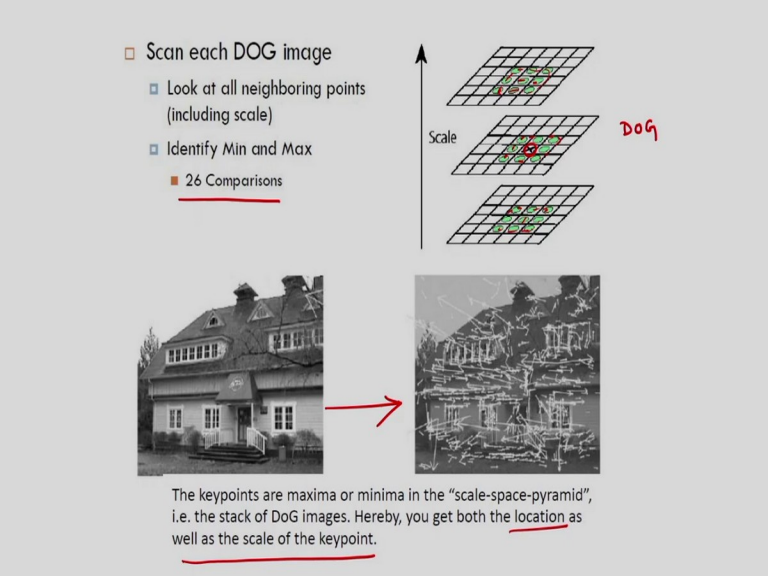
After this the next is the key point localization. So, I am getting the difference of Gaussian images, so detect maxima and minima of difference of Gaussian in scale space, so I have to determine the maxima and the minima of difference of Gaussian in scale space, so here I have

showed the scale space and also I have shown the difference of Gaussians images. So, these are the DOG images, the difference of Gaussians images.

And for each maxima or minima we have to determine the location and the scale of the key points. So, which one will be the key points for this we have to do the comparison. So, each point is compared suppose if I consider this point, so each point is compared to its 8 neighbourhood pixels, so here you can see I have the 8 neighbourhood 1, 2, 3, 4, 5, 6, 7, 8, so 8 neighbours the comparison with the 8 neighbours in the current image and 9 neighbours is in the scale above and below.

So, if I consider this 9 neighbours below the image and this 9 neighbours above the image so that means how many comparisons? There will be 26 comparison, so one comparison that is the 8 comparisons with 8 neighbours in the current image and after this 18 comparisons with the images above and the below of the current image. So, that means I am doing 26 comparisons. So, in the next slide, you can see this comparisons how many comparisons I am doing? I am doing 26 comparisons.

(Refer Slide Time: 38:36)



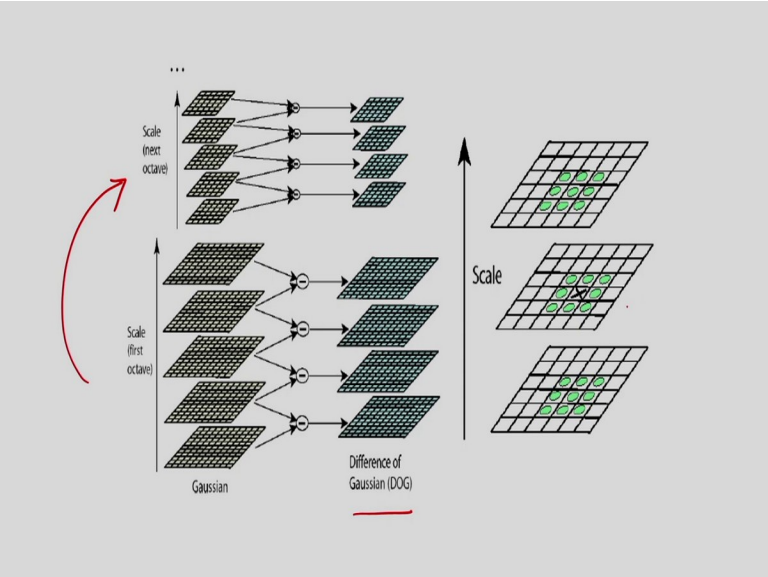
So, here I am showing the difference of Gaussian images DOG, so look at all neighbouring points and we have to see the scale and we have to identify minimum and the maximum so that means suppose corresponding to this point, so each point is compared to its 8 neighbours that

means is point is compared to its 8 neighbours in the current image and 9 neighbours is in the scales above and below that means I have to do the 26 comparisons.

So, comparisons with this all these and comparison with all these points comparison with all these points and based on this I can and determine the key points that is nothing but the maximum value or the minimum points I have to determine from the difference of Gaussian images.

So, corresponding to this input image you can see I am determining the key points the location of the key points. So, for a particular scale I can get the key points. So, that means I will be getting both location as well as the scale of the key points by this process. So, this is the step number one. So, I have to find the approximate location of the key points.

(Refer Slide Time: 40:06)

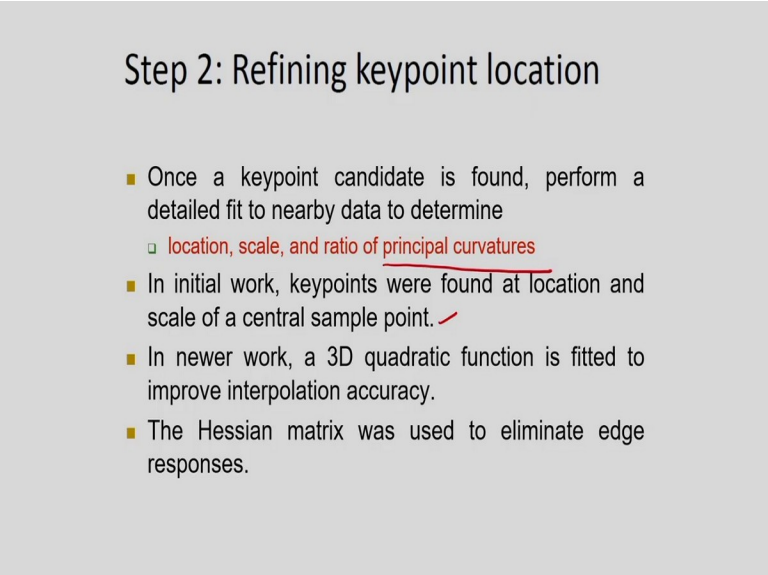


The same thing here again I am showing the same concept already I have explained, so I will be considering the scale space and we can determine the difference of Gaussian and we have to do the down sampling like this, so I will be getting the next octave and after this from the difference of Gaussian, I can determine the location of the key points.

(Refer Slide Time: 40:28)

So, in this image I have shown the initial detection of the key points. So, these are the key points you can see.

(Refer Slide Time: 40:36)



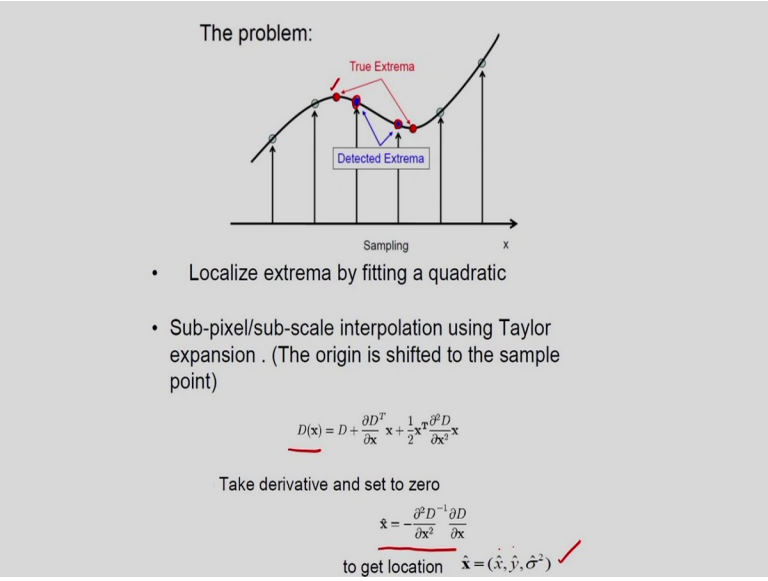
The next one is refining key point location. So, we have determined the approximate location of the key points and in this case we have to perform a detailed fit to nearby data to determine location scale and the ratio of the principal curvatures. So, some of the key points that we have determined in the first step may not be actual key point, so that means we have to do some

refinement, so this refinement we have to do so for this we have to perform a detailed fit to nearby data to determine location scale and ratio of principal curvatures.

So, in initial work key points where found at location and scale of a central sample point, but in the resend work a 3D quadratic function is fitted to improve the interpolation accuracy. So, that concept I am going to explain in the next slide. And also the Hessian matrix is used to eliminate edge responses.

Because sometimes the edge pixels may be detected as key points, but we have to discard this points the edge pixels we have to discard. So, there we have to discard based on the concept of principal curvatures. So, based on this measure I have to determine the principal curvatures and based on this I can neglect or I can delete the edge pixels which are detected as key points.

(Refer Slide Time: 42:04)

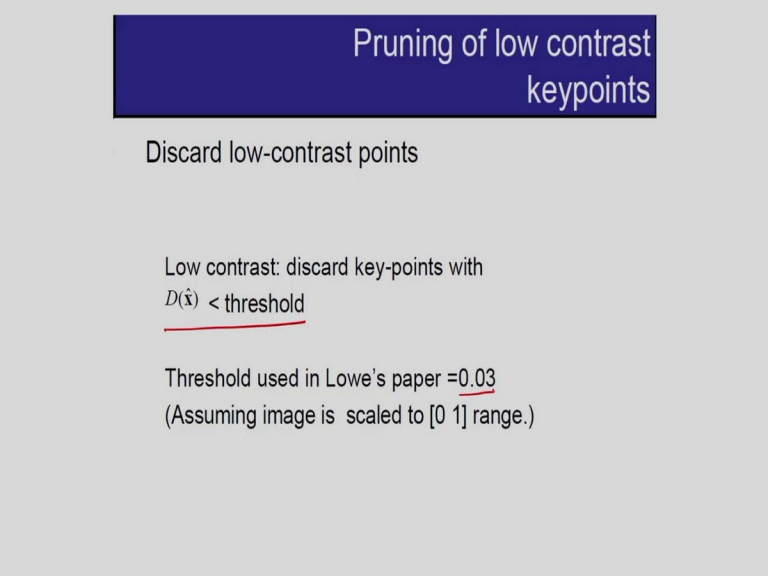


So, how to do the interpolation? Here you can see we have detected the extrema by this comparison does 26 comparisons we have determined extrema so we have determined the extrema but the true extrema are like this the red points these are true extrema, so this localized extrema by fitting a quadratic, sub-pixel or the subscale interpolation is done using Taylor expansion, so you can see the difference of Gaussian is approximated by the Taylor series expansion.

After this we have to take the derivative and set it to 0, so corresponding to this I will be getting the location of the key points. So, I will be getting the location of the key points that is x hat will

be getting, so here you can see I am getting the location x and y and also I will be getting the information above the scale. So, that information is available in the key point definition. So, I will be the coordinates as well as the scales. So, by using this a Taylor series interpolation I can get the actual location of the key points.

(Refer Slide Time: 43:17)

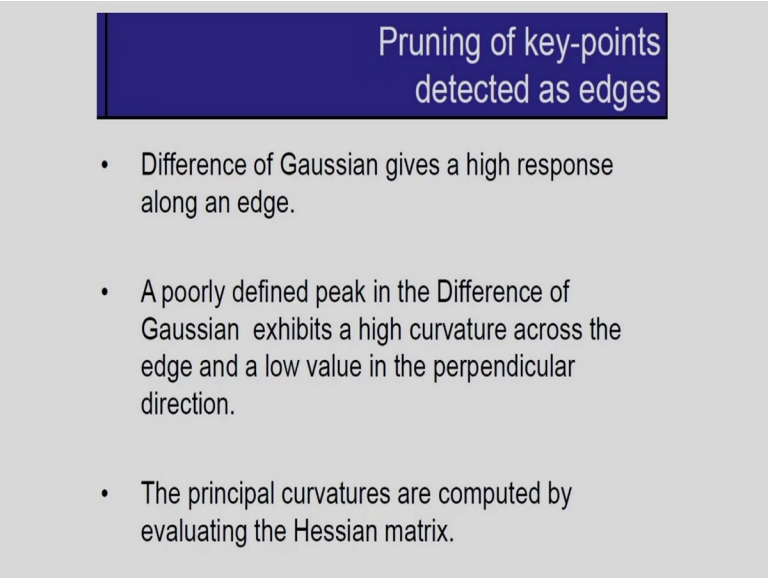


After this we have to discard the low contrast points, so low contrast points are discarded by this condition that is the difference of Gaussian for the point x hat is less than threshold a particular threshold then corresponding to this we can discard the low contrast points. So, generally the threshold is 0.03 depth threshold is used to discard a low contrast points. So, that is the first step that is the I have to discard the low contrast point the key points based on this condition.

(Refer Slide Time: 43:52)

So, here you can see I am showing the example, so in the first image I am getting all the key points, after this I am removing the low contrast key points, so I will be getting the second image that is the low contrast key points are removed.

(Refer Slide Time: 44:07)

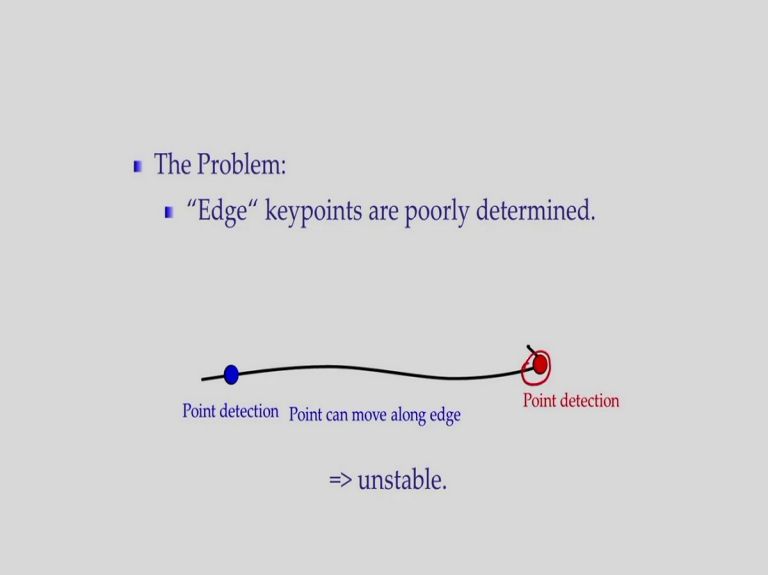


After this I have to consider the edge pixels which are detected as key points. So, we have to neglect this key points that means the edge pixels which are detected as key points should be neglected should be discarded, so for this we are considering one technique that means we have to determine the curvature response we have to determine, so here you can see difference of

Gaussian gives a high response along an edge, so that is why a poorly defined peak in the difference of Gaussian exhibits a high curvature across the edge and low value in the perpendicular direction.

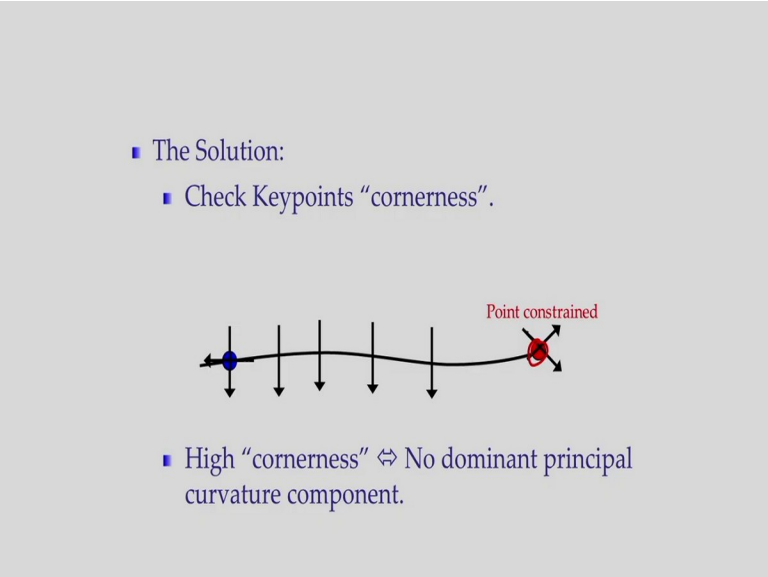
So, the corresponding to the edge I will be getting high curvature across the edge and the low value in the perpendicular direction that is corresponding to the edge points. And this principle curvature is computed by considering the Hessian matrix, so based on the principle curvature I can determine which pixel is the edge pixel or the not the edge pixels, because I have to discard the edge pixels based on the curvature the principal curvature.

(Refer Slide Time: 45:25)



The problem is the edge key points are poorly determine, so what are the key points already we have determined in that case the edge pixels may be present, so that we have to discard based on the principle curvature, so here you can see the point detection point can move along the edge so I am showing the edge pixels and also I am showing the corner points. So, this is the corner points and this is these are the edge pixels I am showing.

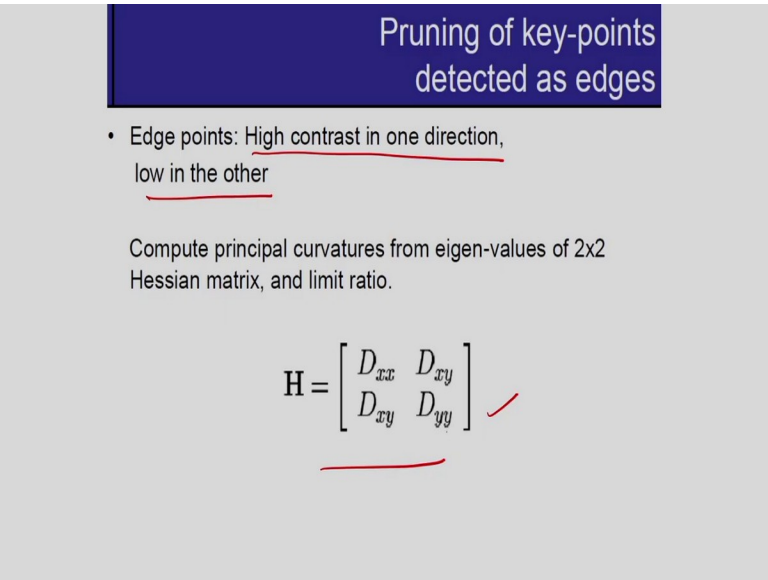
(Refer Slide Time: 45:56)



So, we have to check key points that means the cornerness we have to see, so what is the meaning of this? The high cornerness that is the corner points, no dominant principal curvature component. So, we have to see the high cornerness that means the corner points, so no dominant principal curvature component for the corner points, but if I consider a edge pixels in one direction high curvature, but in the perpendicular direction low curvature corresponding to the edge pixels.

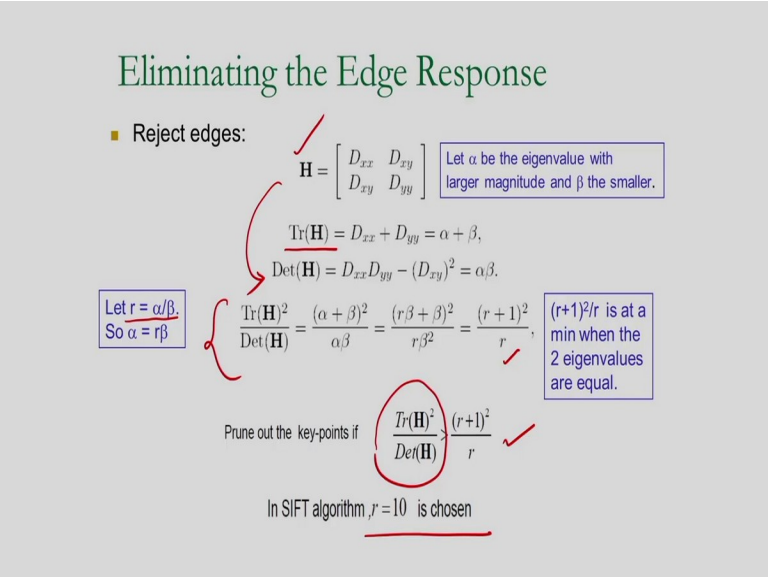
But corresponding to this corner points no dominant principal curvature component. So, by considering the Hessian matrix, we can determine this curvature components, so corresponding to the corner points, no dominant principal curvature component, but the corresponding to the edge pixels in one direction the curvature is prominent and in the perpendicular direction it is low. So, based on this principle we can discard the key points, which are detected as key point.

(Refer Slide Time: 47:02)



So, that means corresponding to the edge point high contrast in one direction and low in the other. So, based on this we can determine the edge pixels and for this for determination of the principal curvature we can consider the Hessian matrix, so we have to compute principal curvatures from eigenvalues of 2 by 2 Hessian matrix. So, this is the Hessian matrix we can obtain from Dxx Dxy Dxy Dyy that already explained in my corner detection class.

(Refer Slide Time: 47:42)



Now, I have to reject the edges, so eliminating the edge response we have to do, so I am considering the Hessian matrix, so let alpha be the eigenvalue with larger magnitude and beta the

smaller eigenvalue. So, corresponding to this Hessian matrix we can determine that trace of H and also we can determine the determinant of H that we can determine in terms of alpha and beta alpha and beta are the eigenvalues.

After this we can determine this ratio that is trace H square divided by determinant H that is nothing but r plus 1 whole square divided by r, so what is r? r is nothing but alpha divided by beta, so that is the definition of r, so that alpha is equal to r into beta. So, r plus 1 whole square divided by r is at a minimum when the two eigenvalues are equal. So, that is not true for the edge pixels.

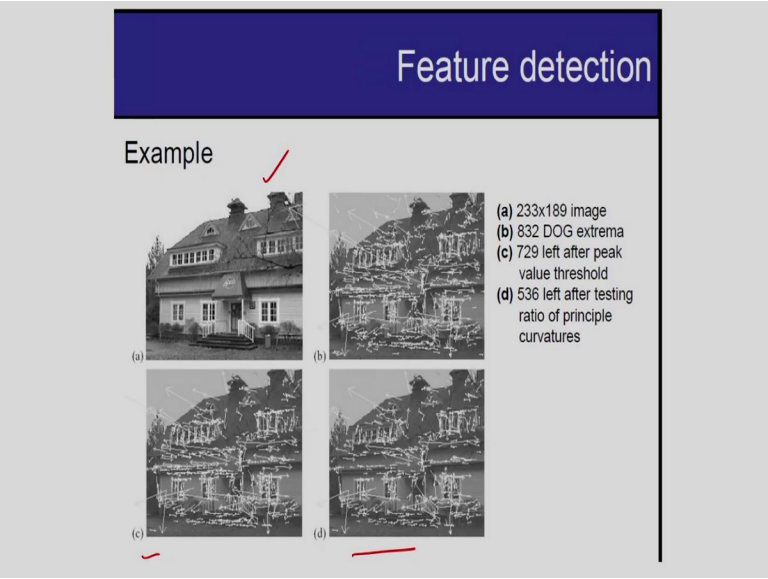
So, for the edge pixel in one direction it is high and in another direction it is low, if this ratio is greater then r plus 1 whole square divided by r, so based on this condition I can neglect the key points corresponding to the edge pixels that means the edge response I am eliminating. So, in the SIFT algorithm r is equal to 10 was considered. So, that means by considering this Hessian matrix I am considering the curvature, the principal curvature and by considering this condition we can neglect the edge response. So, we can eliminate the is response, so that is the concept.

(Refer slide time: 49:29)

So, here you can see removal of high contrast key points residing on edges, so that means the edge pixels which are considered as key points that we can neglect by the principal curvature condition that is a by considering the Hessian matrix we can determine the eigenvalues and

based on this we can neglect the key points corresponding to the edge pixels. So, here you can see removal of high contrast key points residing on edges that we have done.

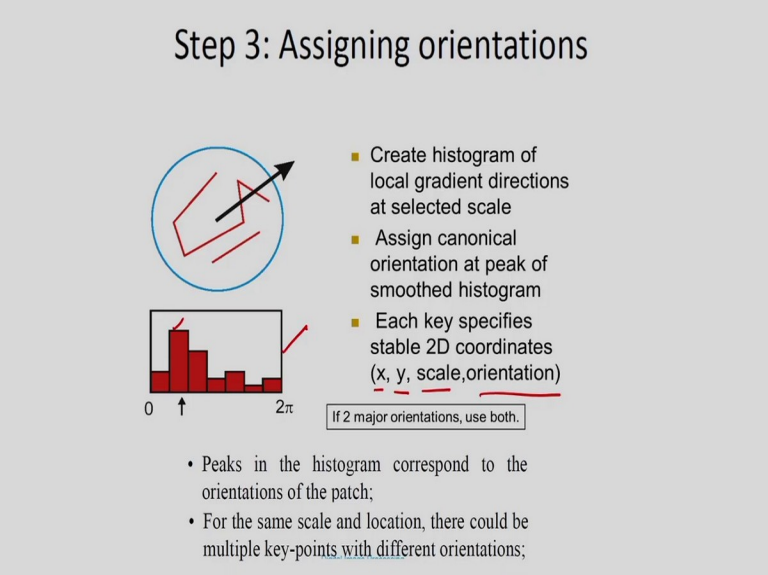
(Refer slide Time: 50:04)



So in this example I have shown the image, so a is the original image the size is 233 into 189, the b is I am considering the a the DOG extrema I am considering, so I will be getting 832 key points that is based on DOG extrema. In the figure c what I am considering? I am getting 729 key points after threshold in the low contrast point.

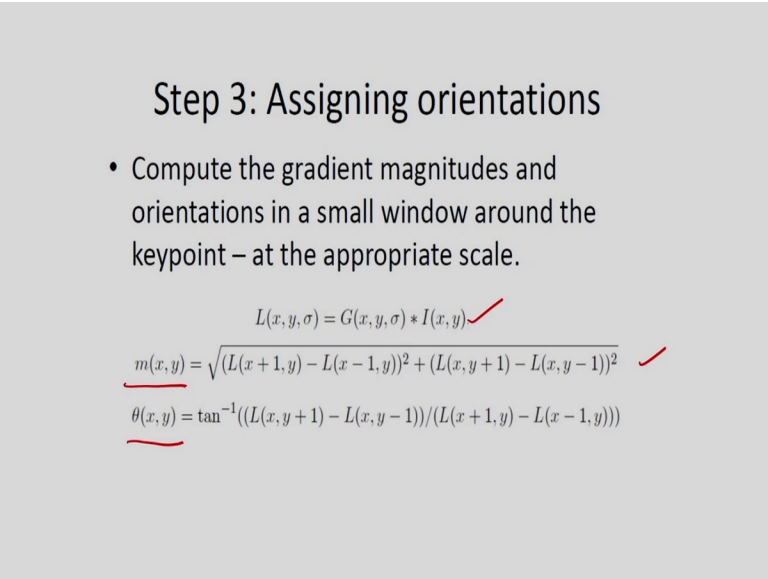
So, low contrast points are eliminated and by this I am getting 729 key points. After this in the figure d that figure, so I will be getting 536 key points after testing the principal curvature that means I am neglecting the key points corresponding to the edge pixels. Finally, I am getting 536 key points, I am getting in the figure d.

(Refer slide Time: 51:00)



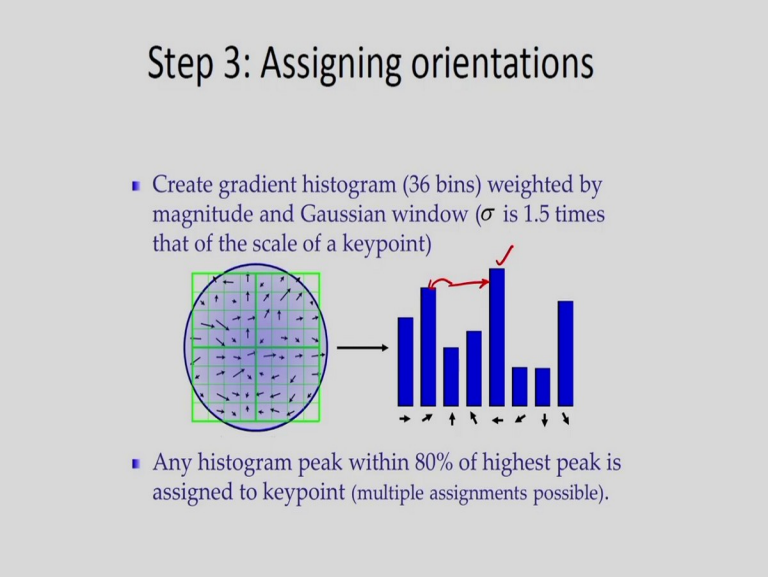
In step number three assigning orientations, so create histogram of local gradient directions at selected scale, assign canonical orientation at peak of smooth histogram, each key specified stable 2D coordinates, so that means the key point corresponds to the coordinate x and y and the scale is available and the orientation is available and a peaks in the histogram corresponds to the orientation of the patches, so I am showing the histogram here and if I considered the peaks it corresponds to the orientation of the patches for the same scale and location there could be multiple key points with different orientations. So, that means I am assigning orientations.

(Refer Slide Time: 51:58)



So, compute the gradient magnitude and orientation in a small window around the key points at the appropriate scale, so here you can see first I am doing the convolution of the image with a Gaussian, so that the image will be smooth, after this I am determining the gradient magnitude that is the gradient magnitude I am determining and also I am determining the orientations. So, by using these equations I can determine the gradient magnitude in the orientations.

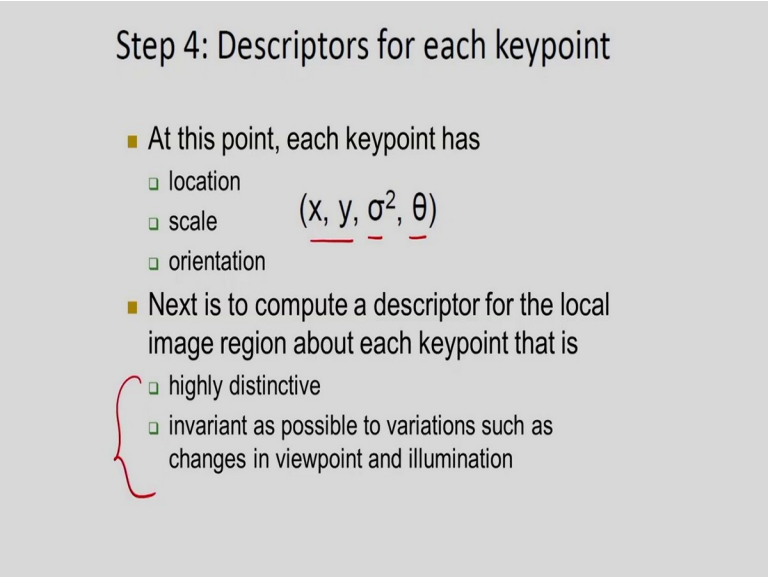
(Refer Slide Time: 52:28)



And after this create gradient histogram, so for this we are considering 36 bins weighted by magnitude and Gaussian windows. So, sigma is 1.5 times that of the scale of a key points. So, I am considering 36 bins and I am creating the gradient histogram. So, any histogram peak within 80 percent of highest peak is assigned to key point that means the multiple assignment is possible.

So, suppose if I consider this is the peak of the histogram, so suppose this is within 80 percent suppose this peak is within 80 percent of this then we have to assign to a key point, so any histogram peak within 80 percent of the highest peak is assign to key points that means the multiple assignments are possible.

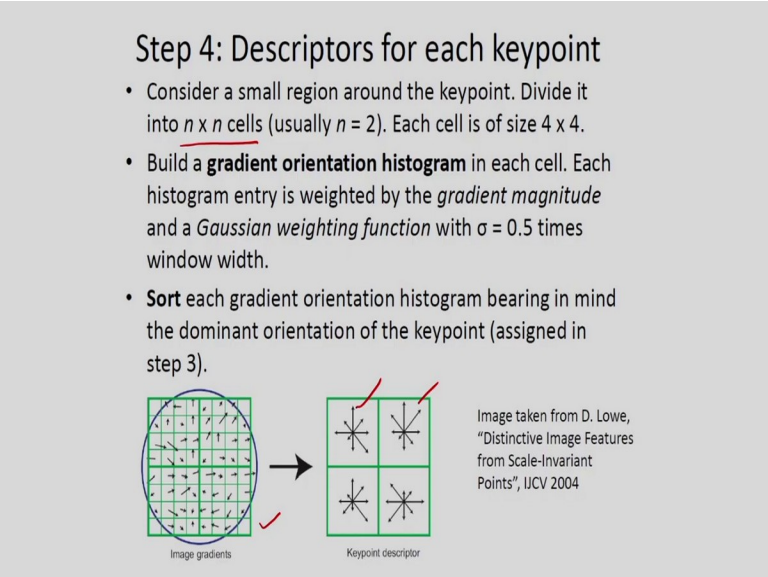
(Refer Slide Time: 53:20)



So, finally in the step number four, we have to find the descriptor for each of the key points. So, the key points are represented like this the x and y coordinates after did the scale the scale is nothing but the variance of the Gaussian function and after this the theta the theta is the orientation.

And we have to compute a descriptor for the local image region about each key point that is highly distinctive also it is invariant as possible to variation such as changes in viewpoint and illumination. So, this points we are considering, so invariant as possible to variations such as changes in viewpoint and illumination.

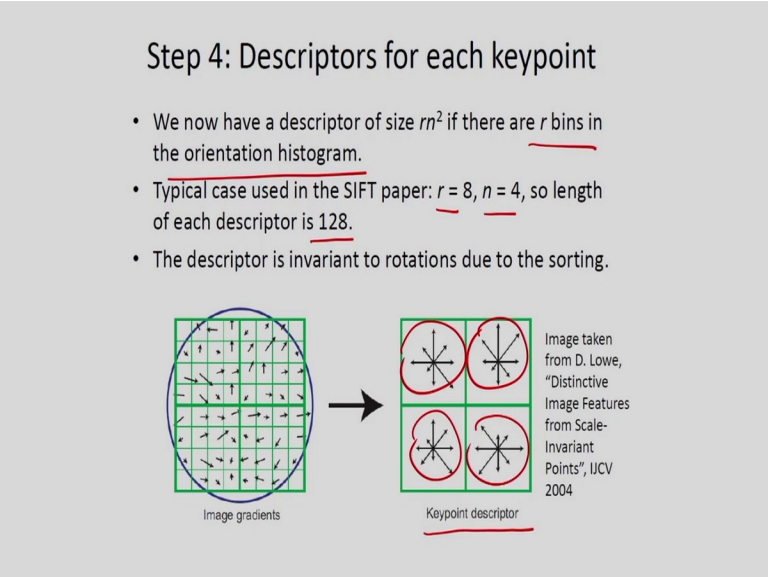
(Refer Slide Time: 54:07)



So, descriptor for each key points, so consider a small region around the key points divide it into n cross n cells, so usually n is equal to 2 and each cell is of size 4 cross 4, after this what we have to consider build a gradient orientation histogram in each cell and each histogram entry is weighted by the gradient magnitude and a Gaussian weighting function with sigma is equal to 0.5 time window width. So, that means I am building a gradient orientation histogram.

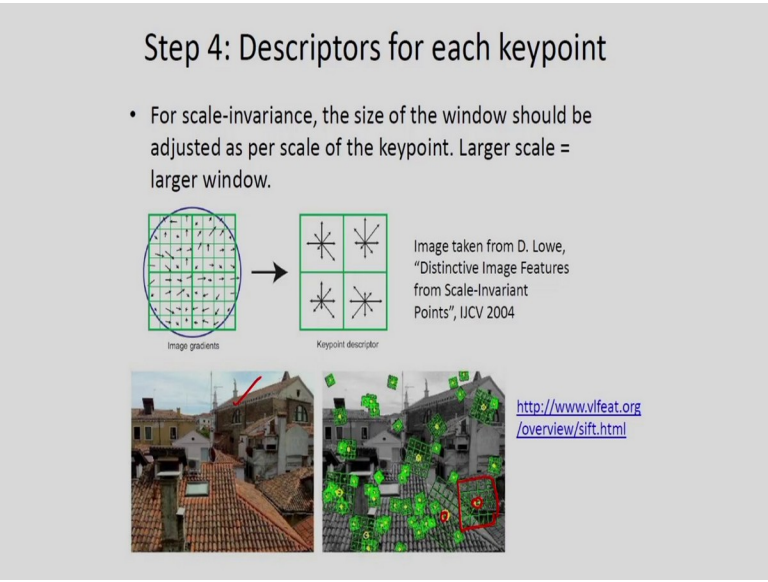
After this sorting is gradient orientation histogram bearing in mind the dominant orientation of the key points. So, here you can see I am showing the image gradients and after this I am considering the gradient orientation histogram in each cell, so I am considering the cells like this, so that means I am building a gradient orientation histogram in each cell and after this I am doing the sorting of gradient orientation histogram. So, if I do the sorting then it will be invariant to rotation. So, that is the objective of the sorting.

(Refer Slide Time: 55:22)



So, here you can see, so we now have the descriptor of size r into n square that means r bins in the orientation histogram and in the SIFT paper r is equal to 8, n is equal to 4, so the length of the SIFT descriptor is 128 and the descriptor is invariant to rotations due to the sorting, so I will be getting the key point descriptors. So, these are cells and I am showing the orientations.

(Refer Slide Time: 55:53)

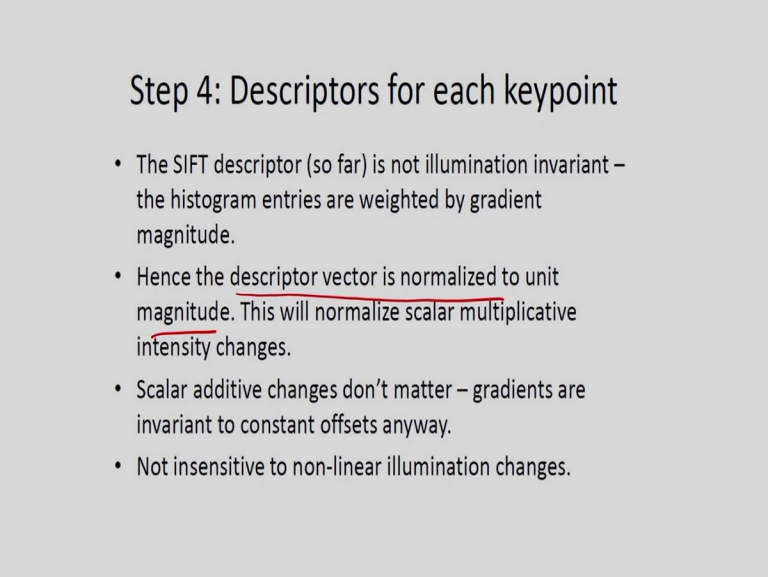


For scale invariance the size of the window should be adjusted as per the scale of the key points. So, larger scale corresponds to larger window, so here in the figure you can see I am showing the

original image I am showing the key points, so suppose this point corresponds to large scale, large scale corresponds to larger window.

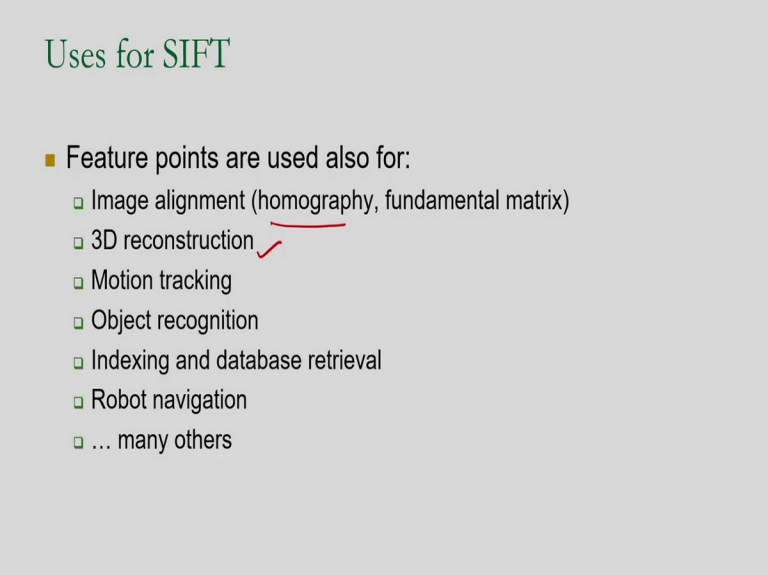
So, that means this larger window I am considering. Similarly, if I consider small scale only the small window we can consider. So, because of this the key point will be scale invariant that is the objective of this, because we have to make the descriptor which is invariant to rotation and the scale. So, I am getting the key point descriptors like this.

(Refer Slide Time: 56:41)



The SIFT descriptor so far is not illumination invariant the histogram entries are weighted by gradient magnitude, so that is why the descriptor vector is normalized to unit magnitude this will normalize scalar multiplicative intensity changes. So, that means the descriptor vector is normalized to unit magnitude and scalar additive changes do not matter because gradient are invariant to constant offset. So, that is why the scalar additive is not a important thing, but we have to see the multiplicative intensity change.

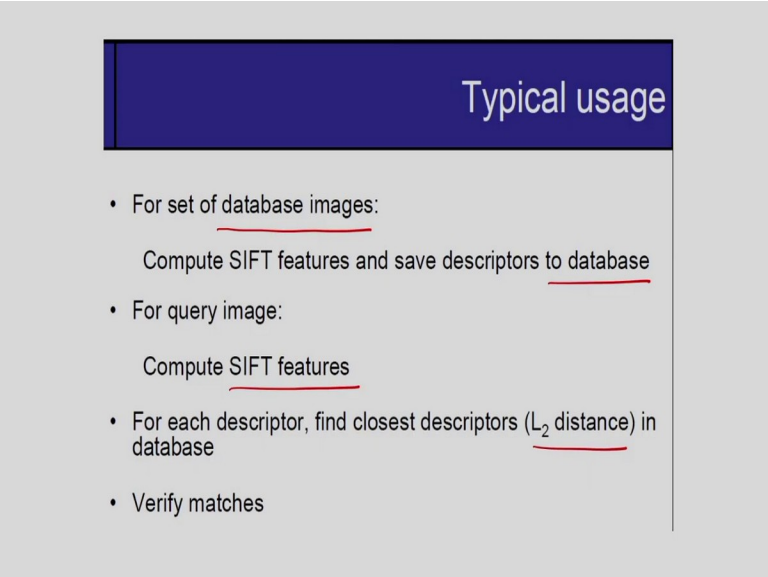
(Refer Slide Time: 57:16)



So, this is about the SIFT descriptors and what are the uses of the SIFT? So, one uses the image alignment like if I consider suppose stereo imaging we can do alignment of the right image and the left image and we can find a correspondence between the images, so we can find the homography we can determine the fundamental matrix for image alignment we can do the feature matching.

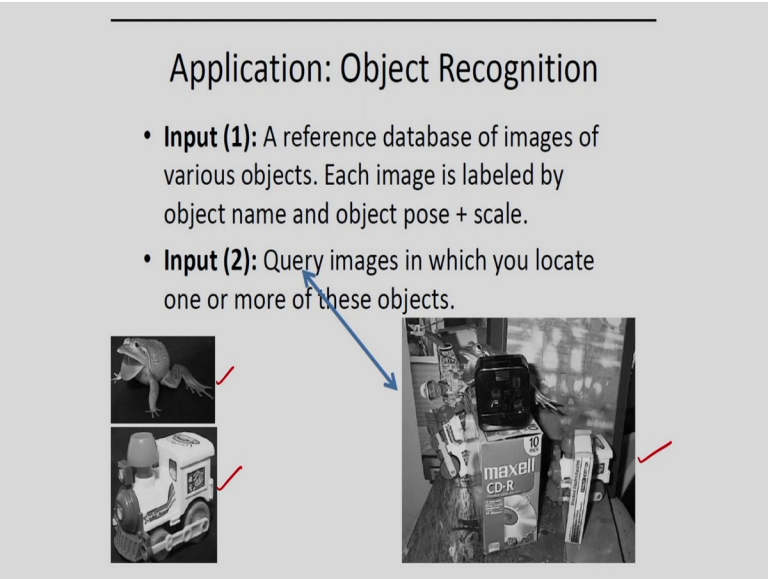
So, by using the SIFT we can do this matching and finally we can do the alignment the image alignment we can do. But a 3D reconstruction also we can extract the SIFT features for motion tracking we can consider a SIFT features, object recognition is one important application in which the SIFT features can be used and indexing and the database retrieval that is nothing but the content-based image retrieval, for this also we can use the SIFT features, robot navigation also we can use the SIFT features and for many other computer vision applications we can use the SIFT features.

(Refer Slide Time: 58:20)



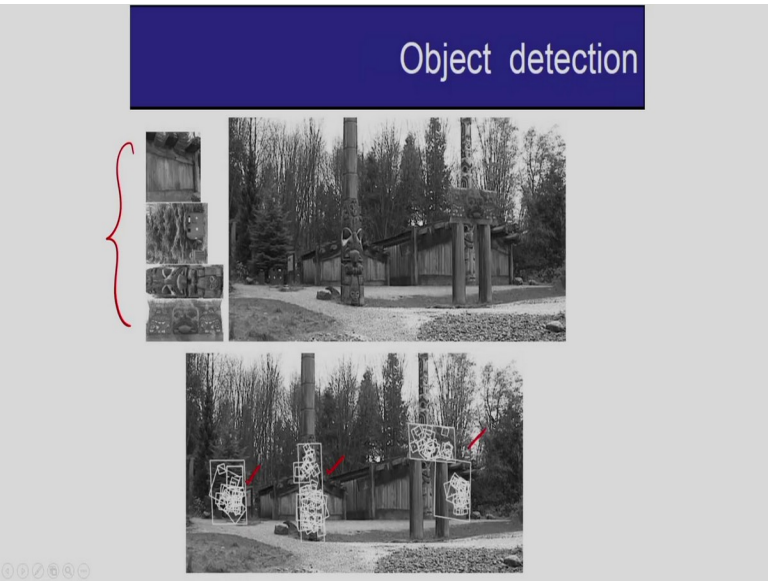
So, one example already I had explained that is the content-based image retrieval. So, for this suppose the database images are available, so for this we can compute the SIFT features and save descriptors to the database and for a query image again we can determine the SIFT features after this we have to compare these features. For each descriptor find closest descriptors in the database. So, for this we can consider L2 distance and based on this we can determine the matching, matching between the test image that is the query image and the image is available in the database that we can do.

(Refer Slide Time: 59:01)



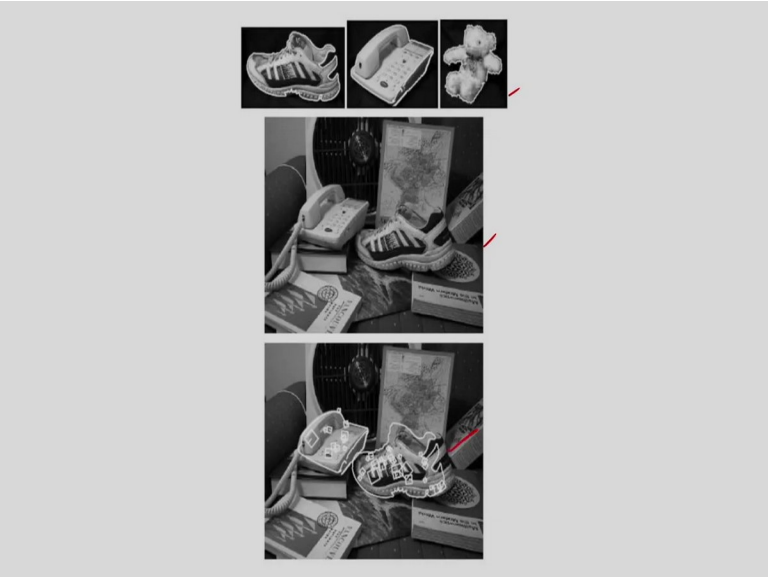
So, here you can see for the object recognition also we can apply the SIFT features, so corresponding to all these objects we can determine the SIFT features and in this input image I can find the corresponding object or any objects based on the matching, so that means again I am explaining so corresponding to these images input images I have to SIFT descriptors and for object recognition again we have to do the matching and based on this matching I can recognize or I can detect the particular object in an image.

(Refer Slide Time: 59:36)



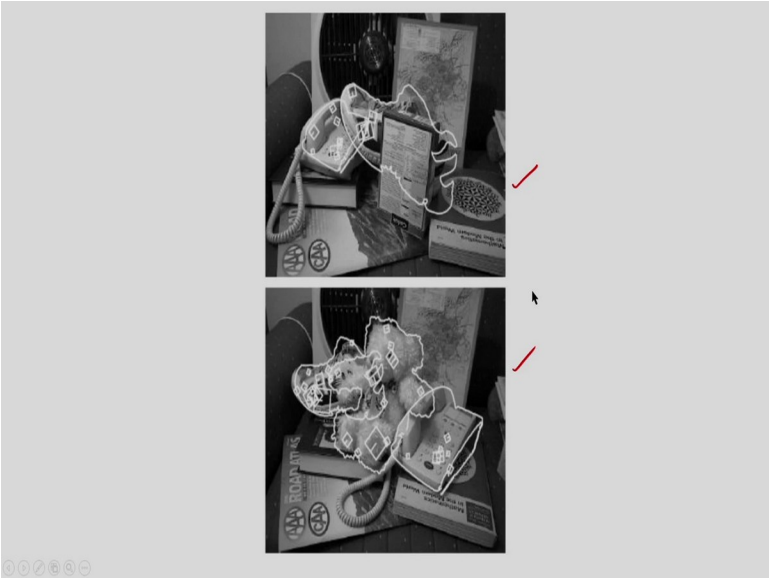
So, like this in object detection corresponding to this images we have the descriptor the SIFT descriptors and we can determine the objects present in an image that is by using the SIFT descriptors.

(Refer Slide Time: 59:48)



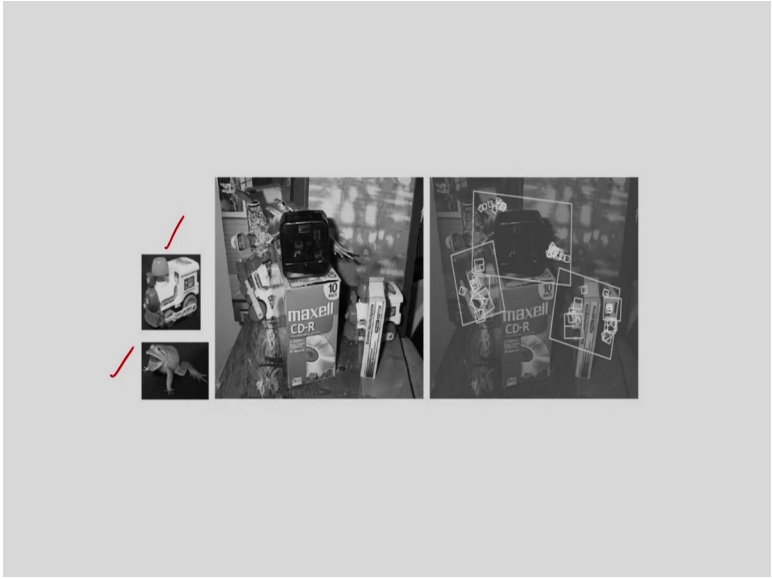
Similarly, I can show these are the input images and for this we have the SIFT descriptors and we can find the objects present in an image.

(Refer Slide Time: 59:58)



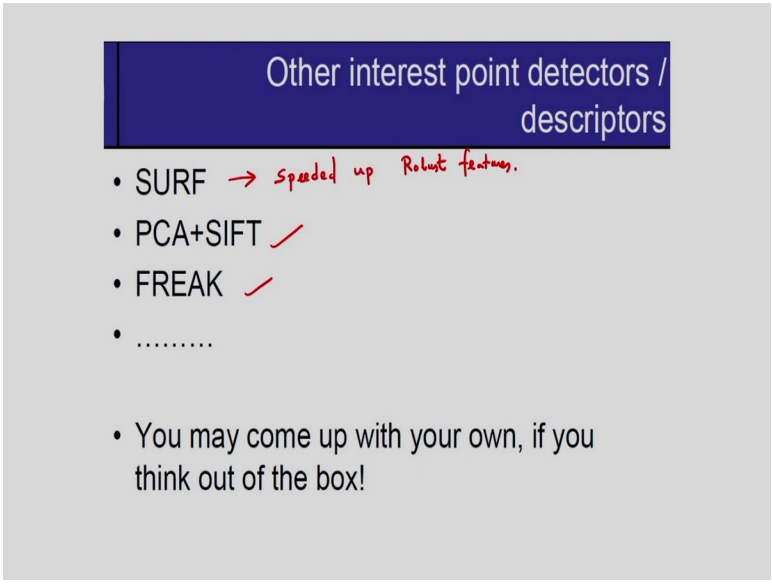
Like this, this is another example, so we can find objects present in the image.

(Refer Slide Time: 60:04)



Again, I am showing the another example. So, these are the input images and for this we have the SIFT descriptors and based on this we can find the objects available in the images. So, this is the concept of the SIFT descriptors.

(Refer Slide Time: 60:14)



And some of the variations of the SIFT are one is the SURF, so SURF is nothing but the speeded up robust feature, so this is also very important that is the speeded up Robust feature, so in case of the SURF the fundamental concept is so much similar to scale invariant feature transformation

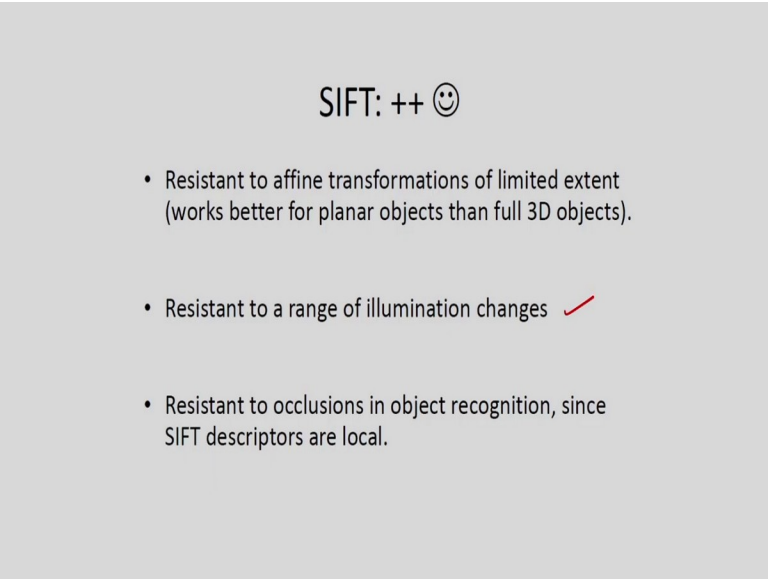
and this SURF is based on Hessian Laplacian operator and SURF shows very similar performance to that of SIFT.

But one advantage of the SURF is faster than SIFT, in case of the SURF the integral image approximation is considered to perform repeat computation of the Hessian matrix and also it is used during the scale space analysis also the difference of Gaussian is used in place of Laplacian of Gaussian for assessing scales. Also, the sum of high wave lets are used in place of the gradient histograms.

So, this depth reduces the dimensionality of the descriptor, which is half depth of the shift. So, these are the difference between the SIFT and the SURF, so one important point is the integral image approach is considered to perform repeated computation of Hessian matrix that is the SURF. So, I am not going to explain the concept of SURF.

But concept is very similar to SIFT, over the difference with the SIFT that I have explained briefly, so one is the integral image approach one is the difference of Gaussian is used in place of Laplacian of Gaussian and also some of high wave lets it's are used in place of gradient histograms. And one is the PCA tipped SIFT one is the FREAK so this you can read from the research papers.

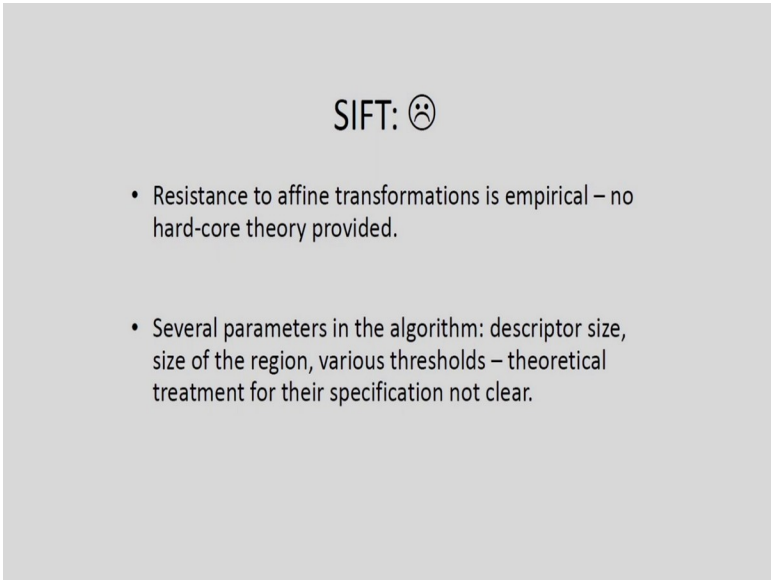
(Refer Slide Time: 62:25)



So, in case of the SIFT what are the advantages? Resistance to affine transformation of limited extent works better for planar objects then full 3D objects and resistance to a range of

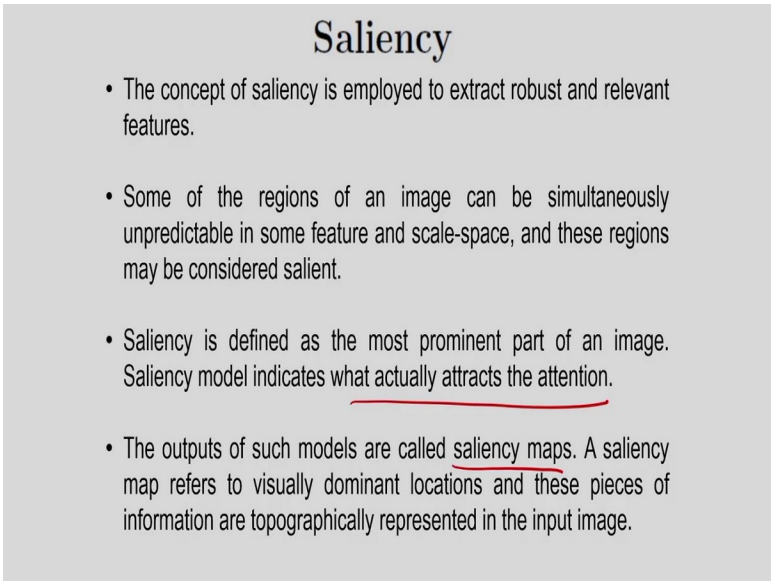
illumination changes this is the advantage resistance to occlusion in object recognition since SIFT descriptors are local, so these are the main advantages of the SIFT.

(Refer Slide Time: 62:48)



And what are the disadvantages? Resistance to affine transformation is empirical no hard-core theory is provided and we have several parameters in the algorithms like the descriptors size, size of the region various thresholds, theoretical treatment for their specifications these are not cleared, so these are the disadvantage of the SIFT.

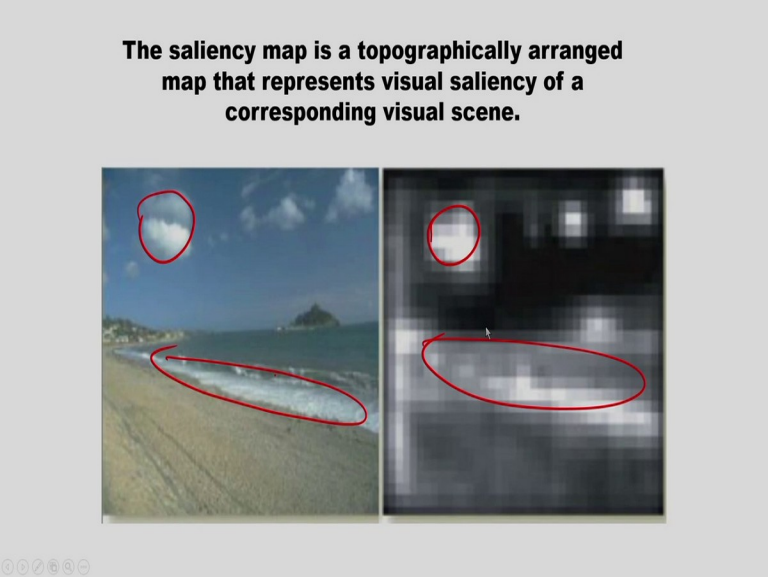
(Refer Slide Time: 63:11)



After this I will explain the briefly the concept of the saliency, the concept of saliency is employed to extract robust and relevant features, some of the regions of an image can be simultaneously unpredictable in some feature and scale space and these regions may be considered as salient, the saliency is defined as the most prominent part of an image and saliency model indicates what actually attracts the attention.

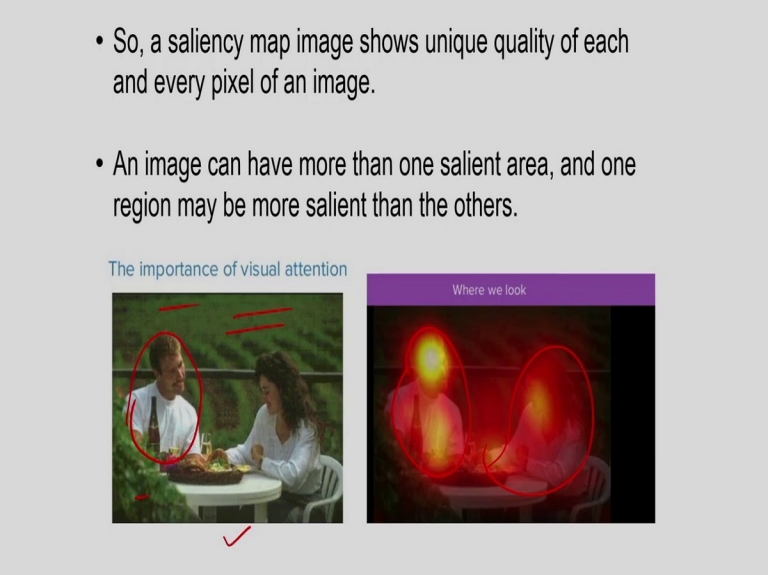
So, that is important. So, what actually attracts the attention that is the concept of the saliency, I will explain in the next slide. And output of such models are called the saliency map. So, from the input image we can determine the saliency map, so the saliency map depends on the attention. A saliency map refers to visually dominant locations and these pieces of information are topographically represented in the input image. So, in my next slide you can see what are the salient regions or the saliency map, I will explain in the next slide.

(Refer Slide Time: 64:21)



So, corresponding to this input image you can see I am determining the saliency regions. So, suppose this region is salient region and similarly suppose these by considered this region this is salient this is the saliency region. So, I will be getting the saliency map that means it indicates what actually attracts the attention that is the visually dominant locations. So, if I consider this regions these are visually dominant locations. So, based on this we can determine the saliency map.

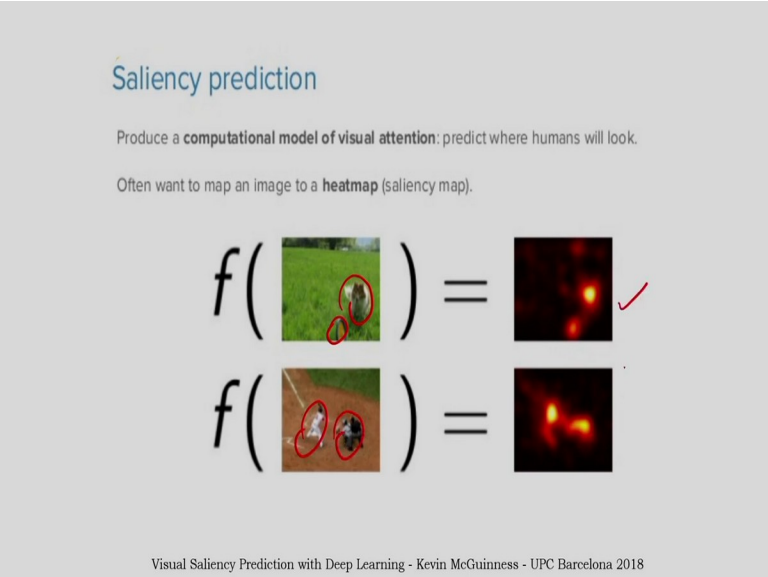
(Refer Slide Time: 64:58)



And so a saliency map image source unique quality of each and every pixel of an image and image can have more than one salient area and one region may have more salient than other regions, so here you can see in this example I am considering these the input image I am determining the saliency map that means it indicates what actually attracts the attention.

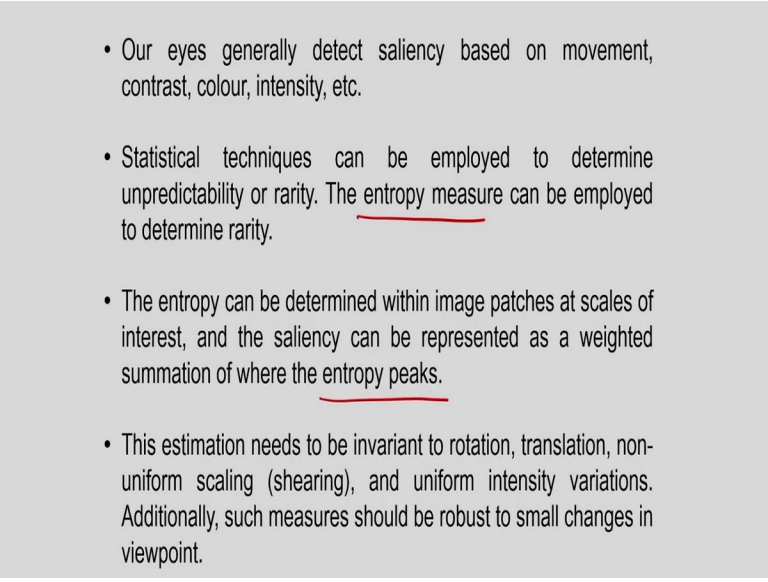
So, that means the visually dominant location. So, if I see so this is the visually dominant location, so you can see I am considering the heat map and similarly this is the visually dominant locations. So, rest of the information is not so important. So, based on this we can determine the saliency map. So, it indicates what actually attracts the attention.

(Refer Slide Time: 65:52)



And you can see we can develop some algorithm for resiliency prediction, so produce a computational model of visual attention, predict where human will look so corresponding to this input image you can see the visually dominant location, so this is the visually dominant locations like this corresponding to this image also visually dominant locations I can determine and from this I can determine the saliency map.

(Refer Slide Time: 66:17)

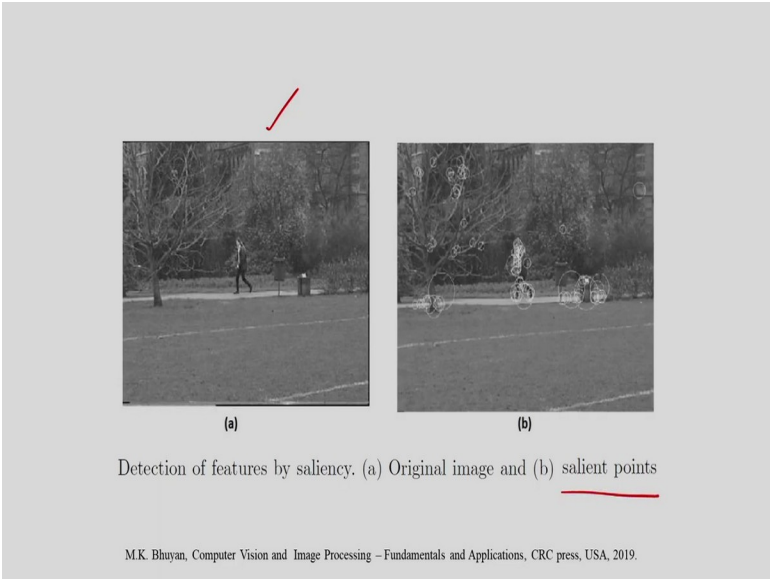


Human eye generally detects saliency based on movement, contrast, colour, intensity, et cetera, so maybe we can employ statistical techniques to determine unpredictability or rarity or maybe

we can employ the entropy measure to determine rarity. So, one is the entropy measure also we can determine to determine the saliency map. The entropy can be determined within image patches at scale of interest and the saliency can be represented as a weighted summation of where the entropy peaks.

So, based on the entropy we can determine the saliency map. The estimation needs to be invariant to rotation, translation, non-uniform, scaling and any from intensity variation and additionally such measures should be robust to small changes in the viewpoint. So, briefly this is the concept of the saliency, so that means again I am repeating the saliency indicates what actually attracts the attention and mainly it is visually dominant locations we have to determine.

(Refer Slide Time: 67:25)



Detection of features by saliency. (a) Original image and (b) salient points

M.K. Bhuyan, Computer Vision and Image Processing – Fundamentals and Applications, CRC press, USA, 2019.

And in this example I have shown the original image and the salient points you can see, so corresponding to this input image I am determining the salient points. So, briefly I have explained the concept of the saliency. So, in this class I discussed the concept of the HOG features that is the histogram of gradient. And after this I discussed the concept of the SIFT the scale-invariant feature transformation, I think for more detail you should see the research papers of HOG the histogram of gradient and also the scale invariance feature transformation. So, let me stop here today. Thank you.