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 22

Edge Detection

Welcome to NPTEL MOOCs course on Computer Vision and Image Processing - Fundamentals and Applications. In my last class I discussed the concept of edge detection, edges and the boundaries are the features of an image. And in that class I discussed how to detect the edges by considering the gradient information. So, I can determine gradient of an image. So, gradient along the x direction and the gradient along the y direction I can determine. And after this I can determine gradient magnitude.

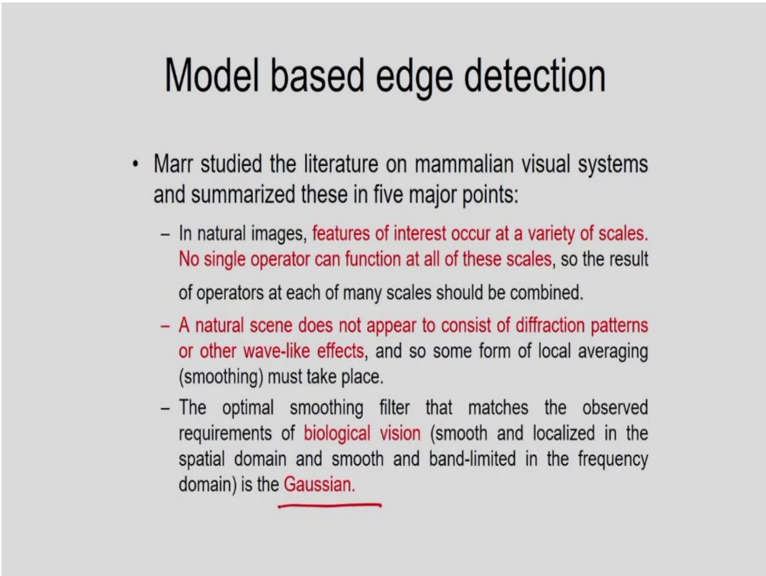
And based on this gradient magnitude, I can decide whether a particular pixel is an edge pixel or not an edge pixel that I can determine. Also I can determine the direction of the edge normal that is the normal to the edge that direction also I can determine that is the angle theta I can determine, that is about the gradient operation. And also I discussed about the second order derivative that is the Laplacian.

So, by using the Laplacian also I can determine the location of the edge pixels. So, for the Laplacian what I have to consider, I have to consider the zero crossings. So, based on the zero crossings, I can decide whether a particular pixel is the edge pixel or not the edge pixel. So, this in last class I discussed about these two techniques.

Today I am going to discuss the edge detection technique and another technique that is the model based edge detection technique. David Marr studied the characteristics of the human visual system that is the mammalian visual system and based on his observations, this model based techniques were developed. So, one technique is the log operation that is the Laplacian of Gaussian that I am going to discuss.

And for this model based technique what we have to do we have to blur the image by a Gaussian function, that is called a Gaussian blurring. And after this I can apply the gradient information that maybe I can consider the first order differentiation I can consider. So, in this case, the maximum will give the location of the edge pixel or I may consider the second order differentiation. So, for this I have to see the zero crossings. So, today I am going to discuss about that model based edge detection techniques. And already I have mentioned and this is mainly based on the observations by David Marr for mammalian visual system that is the biological visual systems. So, let us see what is the model based techniques.

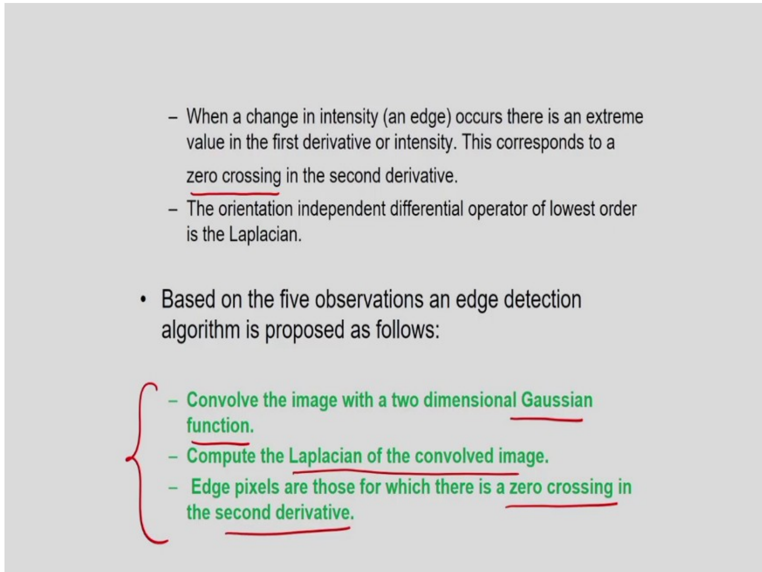
(Refer Slide Time: 3:26)



So, in this case, David Marr studied the concept of the mammalian visual systems and mainly this is the biological vision. So, his first observation is in natural images a features of interest occur at a variety of scales. So, that is why no single operator can function at all of these scales. So, this is the first observation. The second observation is if I consider a natural image, for the natural image, we cannot expect the diffraction patterns or maybe something like the wave like effects.

So, that means some form of local averaging actions must take place. So, that means we have to do some smoothing. And that smoothing I can do by Gaussian blurring, that means the image is convolve with a Gaussian and I can do the smoothing of the image. So, in this case you can see that for smoothing, I can use the Gaussian function that is the Gaussian blurring.

(Refer Slide Time: 4:33)

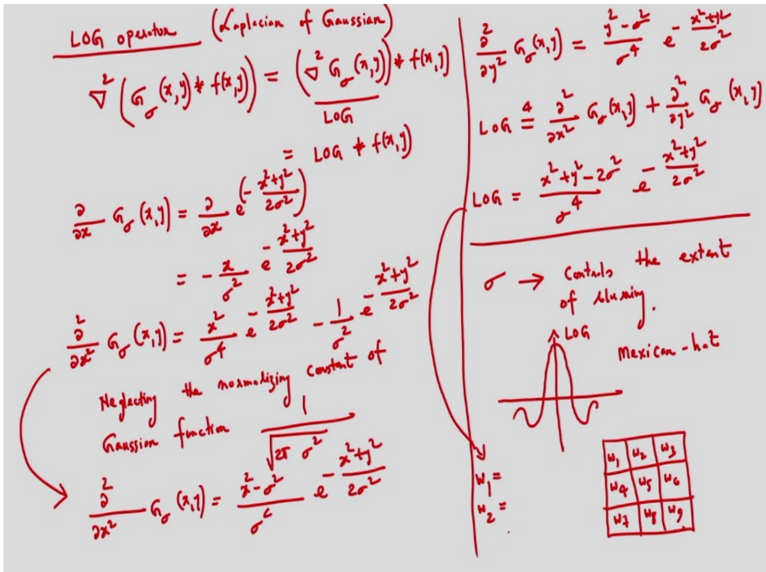


So, his main observations are like this and in this case, based on these observations, you can see, I can apply the gradient operation that is the first order derivative I can consider. And in this case, the maximum will give the location of the edge pixel or maybe I can consider zero crossing also I can consider that is the second order derivative I can consider. And in this case I have to see the zero crossings to find the location of the edge pixels.

So, in case of the model based techniques, what I have to consider, convolve the image with a two dimensional Gaussian function. So, first I have to do the convolution of the image with two dimensional Gaussian function, this is mainly to blur the image, to remove the noises. This is called Gaussian smoothing. So, first one is the blurring the image by a Gaussian function. So, this is the first step. After this I can determine, so, I can compute the Laplacian of the convolve image and based on the edge detection principle what I have to consider, I have to find out zero crossings.

The zero crossings in the second derivative I have to find, that will give the location of the edge pixels. So, that means in the model based techniques, what I have to consider, first I have to go for Gaussian blurring, the image is convolved with a Gaussian function for blurring and after this I can consider the Laplacian operator and for this I have to find a zero crossings that is the model based techniques. So, first I am explaining the concept of the one model based technique that is the log operation, the Laplacian of the Gaussian. So, what is log operation I will explain you.

(Refer Slide Time: 6:26)



So, first one is the log operator, log means the Laplacian of Gaussian. So, what is log operator I can explain, the log operator is the Laplacian of Gaussian, the Gaussian function I am considering to blur the image. So, for this the image is convolved with Gaussian. So, how to write this? Suppose if I consider the Gaussian function, the Gaussian function is G_σ (x, y) and the image is convolved with the Gaussian function I am doing the convolution to blur the image.

And after this I am taking the Laplacian that is the second order derivative I am considering, that is equivalent to you can consider, so suppose G_σ (x, y) I can consider this convolution of these f (x, y). So, if you consider this part only this is nothing but the log the Laplacian of Gaussian you can see it is nothing but a Laplacian of Gaussian. So, I am considering the Gaussian function and after this I am taking the Laplacian of Gaussian. So, this is nothing but the log and after this I am convolving the image with the log operator.

So, Laplacian of Gaussian I am considering and the image is convolved with the log operator. So, what is actually the log operator? So, this log operator I can determine, suppose, if I take the first order differentiation and I am considering the Gaussian function, Gaussian function is σ (x, y). So, I am taking the differentiation along the x direction, $x^2 + y^2\ \sigma^2$ I am considering. So, it will be equal to $\frac{-x}{\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}}$ that is the first order differentiation.

And to compute the second order differentiation, now I am considering the second order differentiation of the Gaussian function, the Gaussian function is G_σ (x, y). So, it will be $\frac{x^2}{\sigma^4}$

$e^{\frac{-x^2+y^2}{2\sigma^2}} \frac{-1}{\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}}$. I am taking the second order derivative and I can neglect the normalizing constant I am neglecting, neglecting normalizing constant of the Gaussian function.

So, that I am neglecting 1 by $\sqrt{2\pi\sigma^2}$ that I am neglecting. Similarly, I can also determine the second order derivative of the Gaussian function corresponding to the variable y. So, what I can determine $\frac{\delta^2}{\delta y^2}$ and my Gaussian function is G_σ (x, y). So, I can determine the Laplacian that is the second order derivative with respect to y. In this case, if I neglect the normalizing constant, then these derivative will be $x^2-\sigma^2$ and $\sigma^4 e^{\frac{-x^2+y^2}{2\sigma^2}}$.

So, this expression I will be getting that is the second order derivative corresponding to the variable x and similarly, for y also I can also determine, so, for y it will be $y^2-\sigma^2$ and it is $\sigma^4 e^{\frac{-x^2+y^2}{2\sigma^2}}$, this is for the second order derivative with respect to y. So, ultimately that log function the Laplacian of Gaussian I can determine that is will be equal to $\frac{\delta^2}{\delta x^2}$ because I have to consider x component and the y component.

So, this is the x component and the y component also I have to consider, that is I have to consider. And finally, the log operator will be, my log operator will be $\frac{x^2+y^2}{2\sigma^2}$ and $\sigma^4 e^{\frac{-x^2+y^2}{2\sigma^2}}$. So, this is my log operator you can see, so, $\frac{x^2+y^2}{2\sigma^2} e^{\frac{-x^2+y^2}{2\sigma^2}}$.

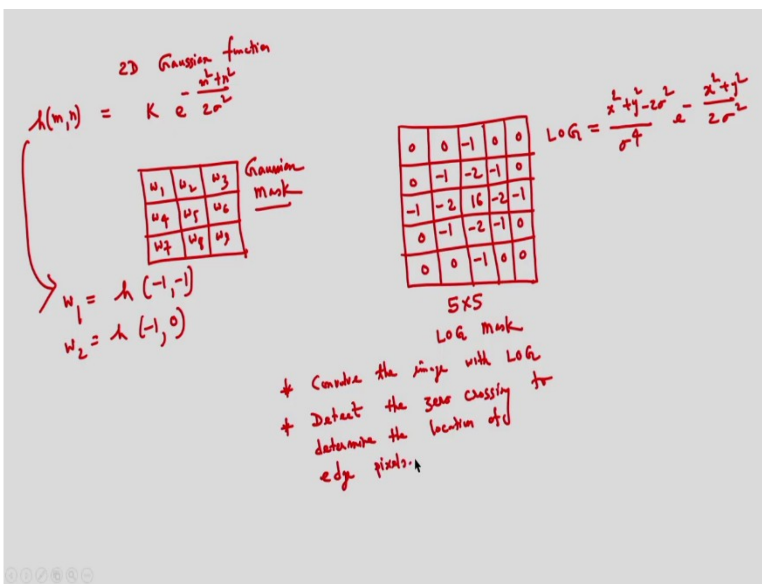
Then in this case I have this parameter, the parameter is sigma, this sigma is a parameter that controls the extent of blurring. So, the sigma controls the extent of blurring, so, you can see the I can get the log operator that is nothing but the Laplacian of Gaussian. So, I am considering one

Gaussian function and after this I am taking the Laplacian. So, you can see I am getting the log operator and this log operator if I want to plot the log operator that is log function, this will be very similar to this shape.

And this is the log function, the log operator, the log function. This shape is something like the Mexican hat shape, Mexican hat shape, this shape so, this is a Laplacian of Gaussian, so shape is something like this. And one important point is the human retina frequency response looks like this. And in this case, it is the bell shape and the sigma controls the extent of blurring. Corresponding to this log operator, I can consider a mask.

So, what will be the mask corresponding to the log operator? Corresponding to the log operator w_1, w_2, these are the weights of the mask. So, these weights I can determine, suppose how to determine w_1? w_1 I can determine from this log function, that log operator I can determine this w value, that is nothing but the discrete approximation of continuous function. So, what I am considering that is corresponding to this log function I am determining the mask and I can determine these the weights w_1, w_2 the all the weights I can determine.

(Refer Slide Time: 15:56)



So, if you see, suppose the Gaussian function is, suppose 2D Gaussian function is something like this, 2D Gaussian function is suppose K e to the power, this is the 2D Gaussian function is suppose h (m, n) $e^{\frac{-m^2+n^2}{\sigma^2}}$. So, corresponding to this Gaussian function you can develop a mask.

What is the mask corresponding to the Gaussian function? So, suppose I am considering the weights w_1 w_2 w_3 corresponding to this Gaussian function I am considering, w_6, w_7, w_8, w_9.

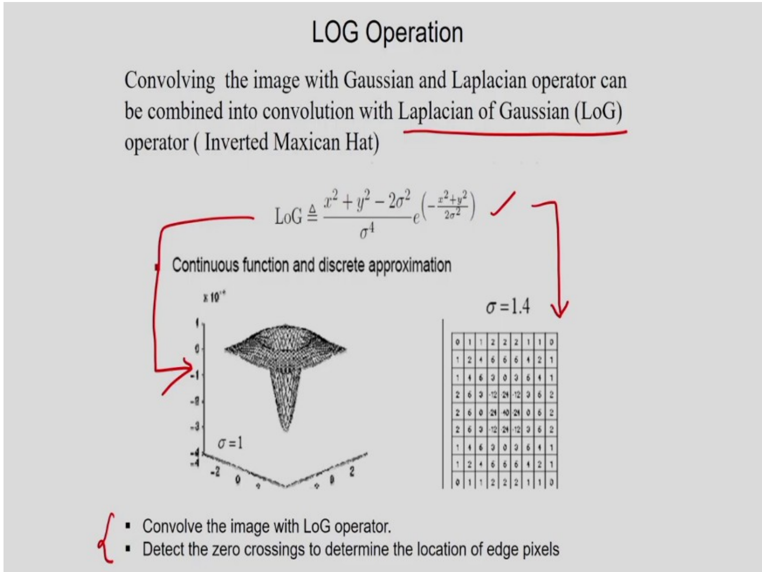
So, what is the weight? The w_1 is nothing but h (-1, - 1). And what is the w_2? w_2 is h (- 1, 0). So, from this you can calculate these values w_1 w_2 like this. So, this is the Gaussian mask. Similarly, I can also show the mask corresponding to the log operation, the log operator. So, let us consider one 5 by 5 mask corresponding to the log function the Laplacian of Gaussian. So, what will be the mask corresponding to the log function, that is the Laplacian of Gaussian?

So, the mask may be something like this - 1, this value the remaining value is 0, 0, 0, 0, - 1, - 2, - 1, 0, this is - 1, - 2 16 the central coefficient is 16, - 2, - 1, 0, - 1, - 2, - 1, 0, 0, 0, - 1, 0, 0. So, this is the mask corresponding to the log operator. So, here I am considering the 5 by 5 mask, that is the discrete approximation of the continuous function because the log function is $\frac{x^2+y^2}{-2\sigma^2}$ divided by $\sigma^4 e^{\frac{-x^2+y^2}{2\sigma^2}}$.

So, this is the log mask. So, this is the Laplacian of Gaussian mask and 5 by 5 mask. So, for determining the edges by using the log operation, so, how to do this? The convolve the image with log operator, so, this is the first step. So, convolve the image which log operator. Next what we have to do, at each pixel observe if there is a transition from one sign to another sign in any direction and based on this decide it as an edge pixel.

That means what I am considering, I am actually detecting the zero crossings, detect the zero crossings to determine the location of the edge pixels. So, the first step is I have to do the convolve, the convolve the image with the log operator the Laplacian of Gaussian and after this detect the zero crossings to determine the location of edge pixels. So, by using this procedure, you can determine the location of the edge pixels.

(Refer Slide Time: 20:26)

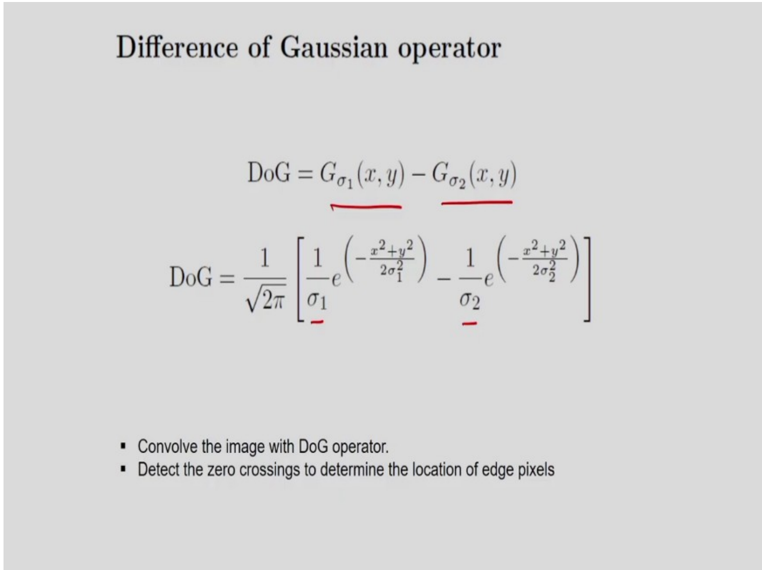


So, here I have shown the log operations. So, convolve the image with Gaussian and the Laplacian operator that is if I combine the Gaussian and the Laplacian operator that is nothing but the Laplacian of Gaussian. So, the convolve the image with the log operator and after this I have to see the zero crossings.

So, already I have explained, so, how to get this the log function that is the log operator, the Laplacian of Gaussian you can determine and corresponding to this log operator you can get the mask something like this if I consider sigma is equal to 1.4, a sigma is the parameter that controls the extent of blurring.

And you can see the shape of the log function, the log function the shape is something like that Mexican hat, so, it is a shape of the log function corresponding to sigma is equal to 1. So, we have to follow these two steps; the convolve the image with log operator and after this we have to detect the zero crossings to determine the location of the edge pixels. So, this is about the log operation.

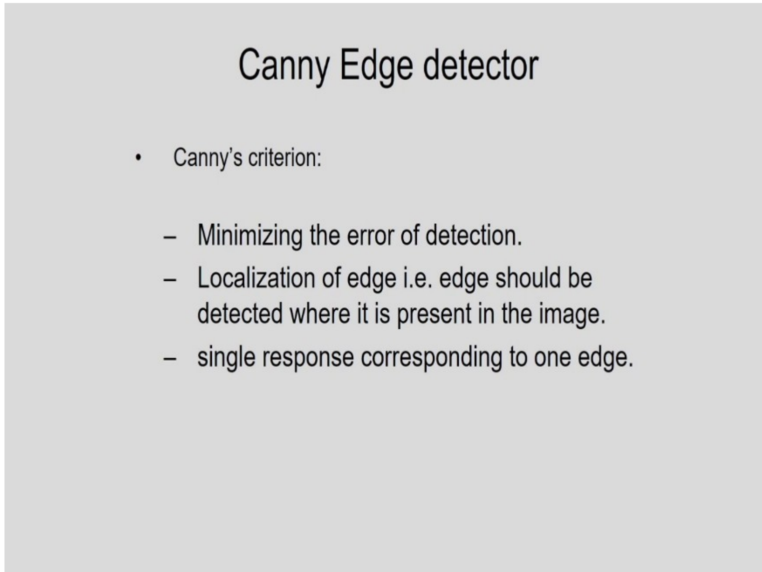
(Refer Slide Time: 21:39)



The another operation I can explain that is the difference of Gaussian operator that is the DoG operation. So, here you can see, I am considering two Gaussian functions, this is the first function and the second Gaussian function and you can find the difference of Gaussians. And I have shown the parameters, the two parameters are there one is σ_1 and another one is σ_2.

So, the procedure is very similar to the log operation. So, convolve the image with the DoG operator the difference of Gaussian and again I have to detect the zero crossings to determine the location of the edge pixels. So, the concept is very similar to log operator, but here I am considering the difference of Gaussians two Gaussians I am considering. So, up till now, I discussed about the concept of the Laplacian of Gaussian and the difference of Gaussian. The next important edge detection technique is the Canny edge detection technique.

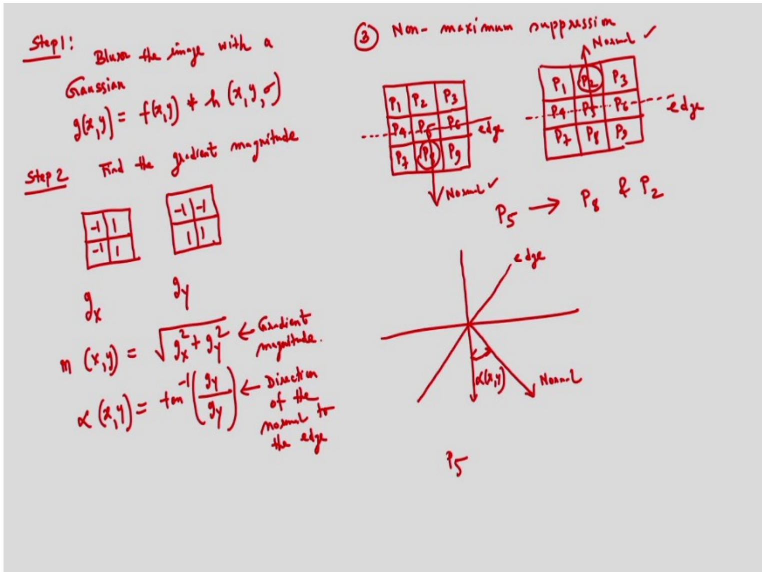
(Refer Slide Time: 22:40)



So, what is the Canny edge detection technique? The Canny edge detector, so the main points are, the first point is we have to minimize the error of detection. So, that means, edge should be detected only when it is present that is the false alarm should be less. Another point is edge should be localized, what is the meaning of this? Edge should be detected where it is present in the image. Another one is, the next concept is the singleness. So, multiple peaks corresponding to single edge point that should be avoided.

So, we have to follow these principles for the Canny edge detection, one is the edge should be detected only when it is present, that is the false alarm should be less. And a second point is edge should be localized, so, that means that we have to detect the edges where it is present in the image. So, edge should be detected where it is present in the image. And the singleness, the singleness means the multiple peaks corresponding to single edge point that should be avoided. So, now, I will discuss the main steps of the Canny edge detection technique.

(Refer Slide Time: 24:07)



So, the first step is, the step number 1, so, first I have to blur the image with a Gaussian function, Gaussian. How to do this? So, my output image is g (x, y); my input image is f (x, y) and after this I am convolving the image with a Gaussian function, the Gaussian function is h (x, y, σ) is the parameter that controls the extent of blurring. So, I have to select the sigma, so, this is the step number 1; the blur the image with a Gaussian to remove noises.

After this I have to find the gradient image that is find the gradient magnitude, I have to determine. So, for this I can consider a 3 by 3 window or maybe I can consider 2 by 2 window. Suppose, if I consider this type of window also I can consider to determine the horizontal gradient and also the vertical gradient I can consider. So, this type of mask also I can consider or maybe the 3 by 3 mask that also I can consider to determine the gradient along the x direction and the gradient along the y direction and based on this I can determine gradient magnitude.

The gradient magnitude is equal to g $(x^2 ¿+g(y^2)$ that I can determine and also I can determine the angle the angle is α ¿x, y); this angle is nothing but the direction of the normal to the edge. So, that angle I can determine the α ¿x, y) I can determine. So, for all the pixels of the image, I can determine the m (x, y) that is a gradient magnitude I can determine and also I can determine the angle α ¿x, y)for all the pixels of the image.

So, one is the gradient magnitude, this is nothing but the gradient magnitude and the α ¿x, y) is nothing but that is the direction of the normal to the edge, that is the direction of the normal to the edge, that I can determine. So, this is step number 2, the step number 3 I want to explain that is a very important step, the step number 3 that is non-maximum suppression, non-maximum. So, the step number 3 is very important that non-maximum suppression.

So, what we have to consider in a window I have to examine, if the gradient value of the pixel is greater than the neighborhood pixels; if it is true, then they retain the pixel as the edge pixel, otherwise, I have to discard the pixel. So, that concept I am going to explain. So, first I have to consider a window. So, in a window I have to examine if the gradient value of the pixel is greater than the neighborhood pixels.

So, suppose I am considering one window here, this is a 3 by 3 window and suppose the pixels are P1, P2, like this P3, P4, P5, P6, P7, P8, P9; these are the pixels. And in this case, suppose if I consider a horizontal edge passing through the point P5. So, I am considering the horizontal edge like this passing through the point P5 that I am considering. So, corresponding to this edge, I have to find a neighborhood pixel. So, corresponding to this edge, this is the normal to the edge one normal.

And again I am drawing the another figure, the same thing that I am drawing. So, P1, P2, P3, P4, P5, P6, P7, P8, P9. So, the edge, this is the edge passing through the point P5 and I am considering the central pixel, the central pixel is P5. And corresponding to this edge, this is the edge, so this is the edge, another neighborhood pixel will be P2 that is the normal to the edge is this is another normal. So, this normal I am considering and also these normal I am considering. So, based on these normals corresponding to the pixel P5, I have two neighborhood pixels, one is P8, P8 is the neighborhood pixel and another pixel is P2.

So, corresponding to the pixel P5 my neighborhood pixels are one is P8 and another one is P2; P8 and the P2. So, what is the edge normal I can again give one example, suppose, I have the edge, suppose this is my edge, this is the edge. So, corresponding to this edge this is the normal to the edge, so this the normal to the edge and in this case, I am considering this angle that is with respect to the reference axis. So, my reference axis, this is the reference axis.

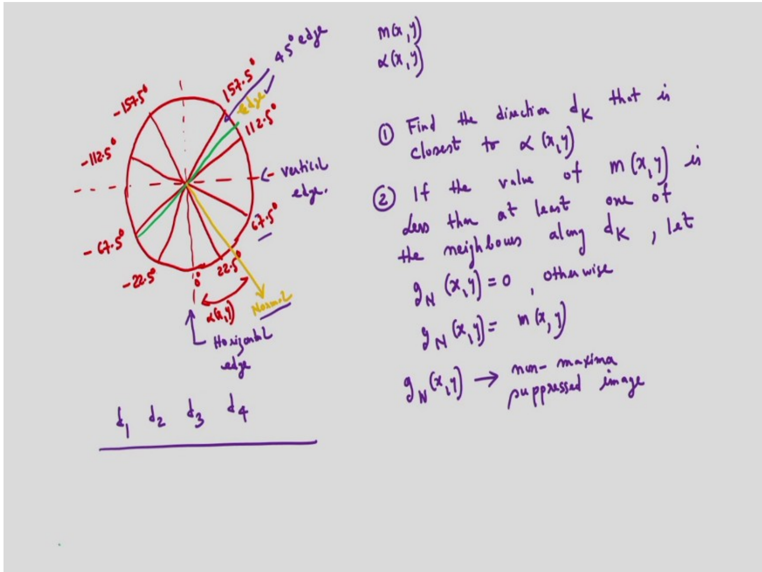
So, this angle is alpha x, y that I am considering. So, here in this case, so, in a window I have to find the neighborhood pixels. So, you can see the edge passes through the point P4, P5, P6 since I am considering the pixel P5 that is the central pixel, corresponding to the pixel P5 my neighborhood pixel will be P2 and the P8 you can see the edge normal, so, I have shown a normal to the edge, this is the one normal and another normal I have shown, a two normals I have shown.

And based on this I can select the neighborhood pixels. So, corresponding to the pixel P5, I have two neighborhood pixels one is P2 and other one is P8. Now, in this case what is the non-maximum suppression? So, in a window I have to examine if the gradient value of the pixel is greater than the neighborhood. So, in this case what I have to consider, I have to compare the gradient magnitude value of the pixel P5 with P8 and the P2.

If it is greater than the gradient magnitude of P8 and the P2 then in this case I have to retain depth as an edge pixel, otherwise, I have to discard P5. So, that means again I am repeating this one first I have to find neighborhood pixels, after finding the neighborhood pixels, what I have to consider? That I have to compare the gradient magnitude of the pixel the pixel is P5 with the neighborhood pixels, the neighborhood pixels are P8 and the P2.

If the gradient magnitude of the pixel P5 is greater than P2 and the P8, then in this case I have to retain the pixel P5 as an edge pixel. If it is not true that means it is less than either P2 or P8, then in this case I have to discard it, that is not the edge pixel. So, this is the non-maximum suppression. So, in this case to find the neighborhood pixels and how to take the decisions, I will explain, now it is not possible to see all the directions to find the neighborhood pixels. So, there is a procedure to do this. So, maybe I can consider only 4 directions to see the neighborhood pixels.

(Refer Slide Time: 34:04)



And for taking the decisions suppose, let us consider this example. In this example, I am only considering 4 directions. So, suppose it is 0 degree, this is 22.5 degree, this is 67.5 degree, this is not in the scale, this is 112.5 degree, this is 157, this is 157.5 degree, this is minus, this side minus 157.5 degree, this is minus 112.5 degree, this is minus 67.5 degree, this is minus 22.5 degree, I am considering.

After this what I am considering, suppose I am considering one edge, suppose my edge is like this, suppose I am considering this is the edge and corresponding to this edge my edge normal will be normal to the edge is suppose, this is the normal to the edge. So, this is my edge and this is the normal to the edge, that is the edge normal. So, this angle is I can consider is α ¿x, y). This angle I can consider as α ¿x, y).

So, this is my horizontal edge and this is the vertical edge and this if I consider this portion, this is 45 degree edge I am considering like this, 45 degree edge. So, you can see here I am only considering 4 directions by defining range. That means, quantization of all possible edge directions into 4 directions by defining the range, I am not considering all the directions to find the neighborhood pixels. Here I am defining the range the range is from 0 to 22.5 degree, that is one range and from 22.5 degree to 67.5 degree that I am considering, so, that is another range.

From 67.5 to 112.5 that is another range, from 112 to 157.5 that is another range. So, just I am defining the range. And this is not in the scale. So, that is what I am doing the quantization of all possible edge directions into 4 directions by defining the range. So, for this I am considering the directions, that directions are d1, d2, d3 and d4. So, 4 directions I am considering and I am considering the 3 by 3 region I am considering.

Now, how to apply the non-maximum suppression? I think you understand this concept, because what is the concept behind this, I am not considering all the directions to find the neighborhood pixels. So, what I am considering the quantization of all possible edge directions into 4 by defining range. So, in this case, I am considering only 4 directions d1, d2, d3, d4 and for this I am defining the range. So, range is from 0 to 22.5, 22.5 to 67.5, 67.5 to 112.5, 112.5 to 157.5.

And in this figure you can see which one is the vertical edge, which one is the horizontal edge you can see. And corresponding to the edge, so, this is my edge and you can see the normal, the normal to the edge you can see and corresponding to the angle α ¿x, y). So, already I have explained that for each and every pixel of the image you have to determine the gradient image that is the m(x, y) you have to determine, also you have to determine α ¿x, y) for all the pixels of the image you have to determine.

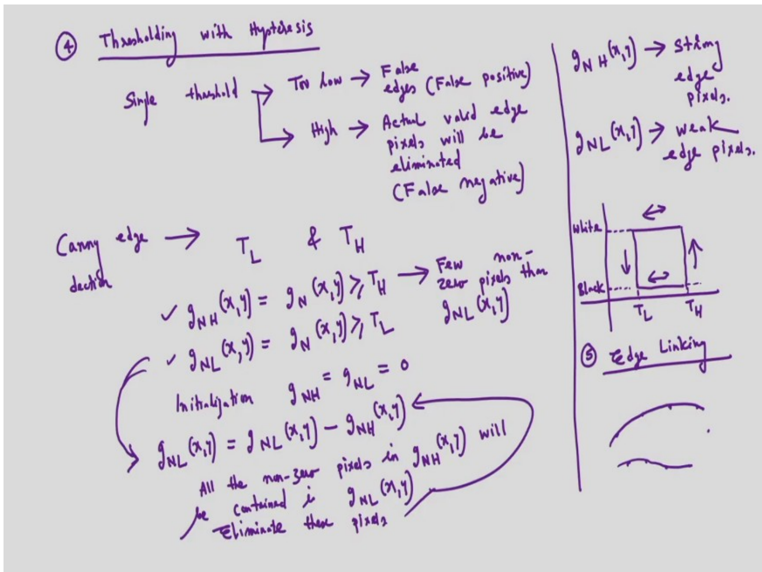
Now, based on this concept, because I have the 4 direction only, so, how to do non-maximum suppression? So, first point, first step what you can do, find a direction dk So, I have only four directions d1, d2, d3, d4; find a direction dk that is closest to the angle α ¿x, y). So, the first I have to find the direction dk, that is closest to α ¿x, y). So, for all the pixels of the image, we have to do this. And after this we can find a neighborhood pixel. So, based on this the condition number 1, based on the step number 1, we can find our neighborhood pixels.

If the value of the gradient magnitude m (x, y) is less than at least one of the neighborhood, at least one of the neighbors along the direction, direction is dk; along the direction dk, then so what I have to do, let gN, so, that gN is the non-maximum suppression image, g_N (x, y) will be equal to 0, that means I am suppressing that one. And otherwise what we have to, otherwise what I have to do, otherwise g_N (x, y) that is I have to retain the pixel that is equal to m (x, y).

So, if the value of the m (x, y) that is the gradient magnitude is less than at least one of the neighbors along the direction dk, then what I have to consider, I have to the suppression, the suppression is nothing but g_N (x, y) will be equal to 0. And otherwise we have to retain that pixel, that is the pixel is x, y is the pixel and corresponding to this g_N (x, y) will be m x, y. So, what is

g_N (x, y) here? g_N (x, y) is non-maximum, non-maxima suppressed image. So, this is a very important step of the Canny edge detection technique that is how to apply the non-maxima suppression. The next step is the step number 4.

(Refer Slide Time: 42:18)



The step number 4 is thresholding with hysteresis. Now in this case because I have to select the edge pixels, so for this I have to consider a threshold for comparison, that means we have to reduce the false edge points. So, for this I have to consider the thresholding operation. So, if I consider only single threshold for this comparison that means whether the pixel is the edge pixel or the not edge pixel, if this value is greater than a particular threshold, based on this I can consider that pixel is the edge or the not the edge pixel.

So, if I consider only the single threshold, so there is a problem. The problem is, if the threshold is too low, the threshold if I select that is the too low threshold value, so what will happen? I will be getting false edges. That is, I can consider as false positive, that is corresponding to the low threshold. And if I select suppose very high threshold, if the threshold is very high, again there will be a problem. So, what is the problem? Actual valid edge pixels will be eliminated.

That is nothing but false negative, this is the false negative. So, that is the problem with the single threshold. So, in the Canny edge method, Canny edge detection, it uses two thresholds, that is one is the low threshold TL, and another threshold is the high threshold that is TH. One is the low threshold; another one is the high threshold it is used. And based on this how to take a decision, you can see, so g_{NH} first I have to determine that is the first I have to do the thresholding g_{NH}, I am getting that is by comparison with the high threshold I am getting g_N (x, y).

It is compared with the high threshold and corresponding to this I am getting g_{NH} x, y I am getting. So, also I can compare with the low threshold and I am getting g_{NL} x, y and if I compare with the another threshold, another threshold is the low threshold, because I have two thresholds in the Canny edge detection. In g_{NH}, in this case I have few, fewer non-zero pixels, than g_{NL} x, y. So, that is obvious because g_{NH} I am obtaining by comparison with the high threshold.

So, in this case I have only few non-zero pixels than g_{NL} (x, y). So, that is you can see how to get these two values, one is the g_{NH}, another one is g_{NL}. These are the images, that is I am comparing with the two thresholds, one is the high threshold, another one is the low thresholds. And for initialization, how to do the initialization? So, initialization what I have to do, both the cases g_{NH} and g_{NL}, the initialize is 0. This is initialize 0 and after this I can determine g_{NH} and the g_{NL} I can determine.

And another step I can consider, the g_{NL} (x, y) is equal to g_{NL} (x, y) from this I am doing this one g_{NL} (x, y) minus g_{NH} (x, y), that I am doing, what is the meaning of this? So, that means all the non-zero pixels in g_{NH} will be contained in g_{NL} (x, y). So, I am writing this one, there is why, that is why all the non-zero pixels in g_{NH} (x, y) will be contained in g_{NL} (x, y). So, that is why I am doing this step, this step means, this step I am doing. That means eliminating these pixels, I can eliminate these pixels.

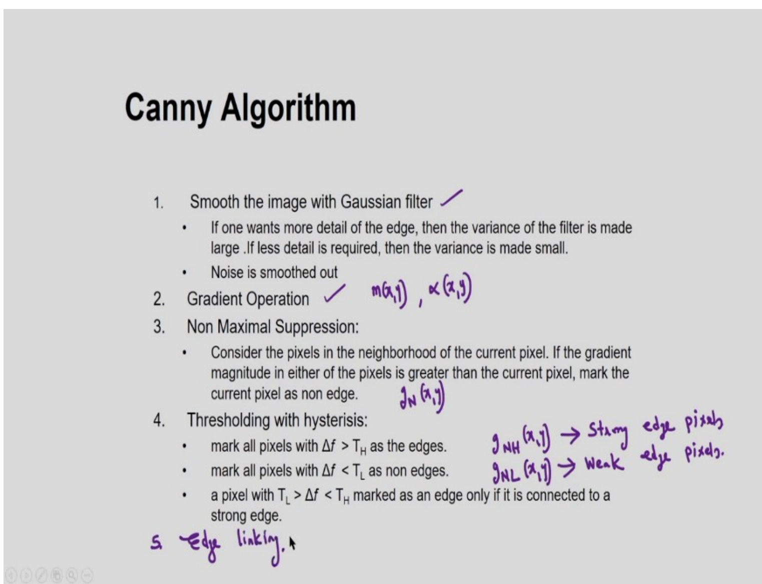
So, eliminate these pixels. So, I can eliminate these pixels by using this operation. So, that means in this case, I am after doing all these operations I will be getting these two images, one is g_{NH} (x, y), that corresponds to the strong edge pixels. Another image I am getting g_{NL} (x, y), that corresponds to the weak edge pixels. So, I am getting the strong edge pixels and the weak edge pixels. The thresholding operation I can draw like this. What I am actually doing?

The thresholding with hysteresis. So, this threshold is the low threshold, this threshold is the high threshold and corresponding to this, this is a black that is in the dark, black and corresponding to this point this is white. Because after edge detection I am getting the binary image, either the pixel is present or not, that is I am getting the binary image after the edge detection technique. So, this is, this I am showing how to do the thresholding with hysteresis.

So, after thresholding with hysteresis, I am getting two images, one is the g_{NH} that corresponds to strong edge pixels and g_{NL} x, y that corresponds to weak edge pixels. Then finally what I have to consider because I have the edge pixels, to get the boundaries I have to edge linking. The final step I can show you, the final step will be the edge linking, so the step number 5 is I can consider the edge linking, I have to connect the pixels.

The edge pixels I have to connect to get the boundary. So, I have the edge pixels like this and I have to connect the edge pixels based on some conditions and to get the edges I have to do this. Because by considering these steps, the step number 1, step number 2, step number 3, step number 4 I can get the strong edge pixels and the weak edge pixels. After this I have to do the edge linking. So, this is about the Canny edge detection technique, the edge linking procedure I will explain.

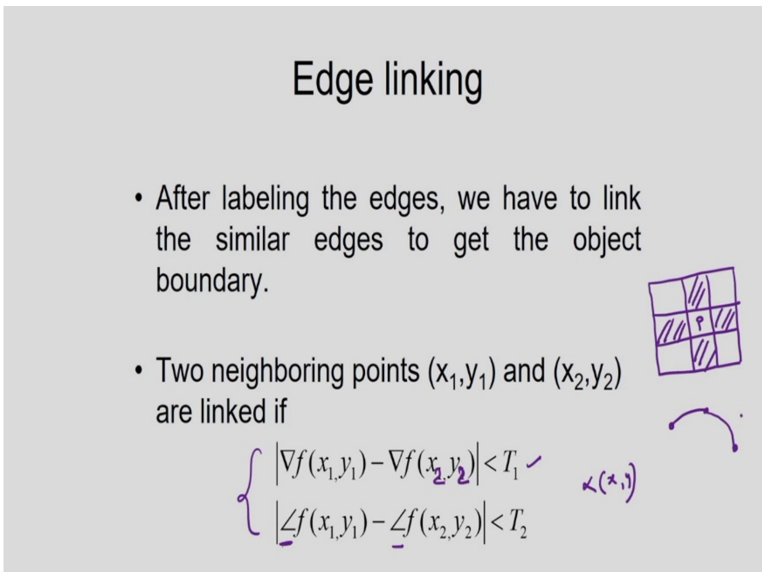
(Refer Slide Time: 51:53)



So, what is the Canny edge detection? So, first I have to go for the Gaussian blurring, this step already I have explained. So, the image is convolved with the Gaussian to remove noise, this is the first step. Second one is, I have to do gradient operation, so for this I have to determine the gradient magnitude image that is the m (x, y). And also the angle also I can determine, angle is α ¿x, y), that is the direction of the edge normal.

After this I have to apply the non-maximum suppression technique and I will be getting the image gN x, y, that is the non-maximum suppressed image I will be getting. So, this procedure already I have explained. So, for this I have to compare the neighborhood pixels and finally what I have to consider, I have to consider two thresholds, one is the low threshold, another one is the high thresholds. And based on this I will be getting two images, one is g_{NH} x, y, that corresponds to strong edge pixels and another image is I am getting g_{NL} x, y, that corresponds to weak edge pixels. So, these are the steps and after doing this, the step number 5 will be the edge linking.

(Refer Slide Time: 53:29)



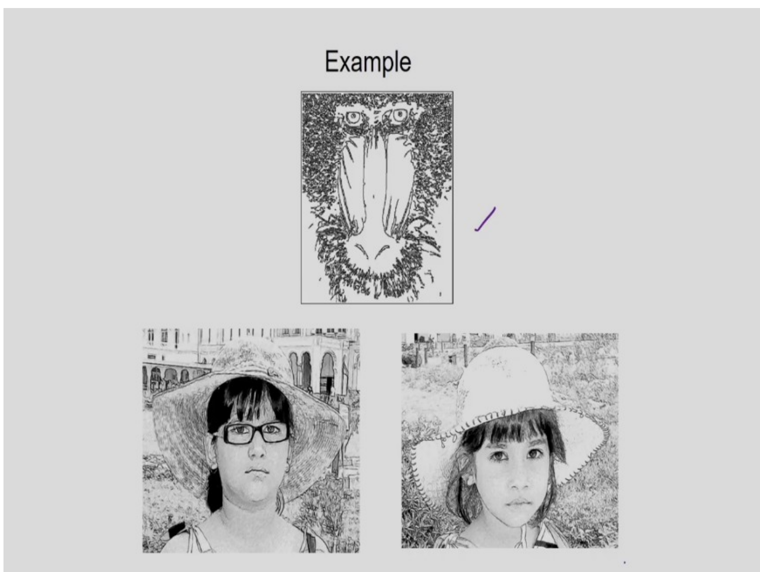
So, you can see, how to do the edge linking you can see. That means we have to give the boundary, so suppose if I consider a 4 connected neighborhood, suppose if I consider one window, 3 by 3 window. And corresponding to this central pixel, the central pixel is suppose P, if I consider the 4 connected neighborhood, so this is my neighborhood pixels corresponding to the 4 connected neighborhood. So, in this case to get the boundaries, what I have to consider?

Corresponding to each edge point, we have search for similar points in the connected neighborhood. So, I am repeating this, because I have to find the boundary, the object boundary I have to find, I have to connect the edge pixels. So, corresponding to each edge point, we search for similar points in the connected neighborhood. So, in this case you can see, these two points I am considering, one point is x_1, y_1; another point it should be actually it should be x_2, y_2.

I am determining the gradient for the point x_1, y_1; f (x_1, y_1) the gradient magnitude I am determining. And also I am determining the gradient magnitude corresponding to the point x_2, y_2. If it is less than a particular threshold, the threshold is T1, that condition I can take. And also if you see the second condition, what I am considering, I am determining the direction, the direction is the angle, this angle I am considering, that means I am determining the angle.

So, f (x_1, y_1) that I am determining and f (x_2, y_2) that is the angle that I am determining, the angle is, already I have defined angle is nothing but α ¿x, y), that is the direction of the edge normal. And if these two conditions are satisfied, then that based on these conditions, I can join the edge pixels, suppose these are the edge pixels, so this pixel and pixel I can join. Similarly, if this condition is satisfied, between these two points then I can join the pixels to get the boundary, to get the edge, the continuous edge. This is the edge linking procedure.

(Refer Slide Time: 55:49)



So, here I am showing one example of the edge detection by Canny edge detection technique. This is the first example and second example also you can seem so we have applied Canny edge detection technique to determine the edges and the boundaries. Up till now, I discussed about the concept of the edge detection by using the model-based techniques. So, first I discussed about the log operation, that is the Laplacian of Gaussian.

So, for this, the image is convolved with Gaussian and after this we can take the Laplacian. So, that means first I have to take the image and after this the image is convolved with the log operator that is the Laplacian of Gaussian. And after this I have to find the zero crossings. And based on the zero crossings, I can decide whether a particular pixel is the edge pixel or not the edge pixel. That is about the log operator.

And similarly, I have also discussed about the DoG operator, that is the difference of Gaussian. After this I considered the Canny edge detection, so there are five steps. The first step is, first I have to blur the image, so convolve the image with Gaussian to blur the image to remove the noises. So, this is the step number 1. Step number 2, I have to find the gradient magnitude in the image, that is m (x, y) I have to determine and also, I have to determine α ¿x, y), that is the direction of the normal to the edge I have to determine. That is the step number 2.

The third step is very important, that is the non-maximum suppression that I have to consider and for this I have to consider neighborhood pixels. So, instead of considering all the directions I can consider only few directions, the quantization of all the directions into the few directions, in my example I have considered only 4 directions and I can find the neighborhood pixels. And after this I can determine the non-maxima suppressed image, the non-maxima suppressed image I can determine because I have to consider the neighborhood pixels.

And finally, I have to apply the thresholding with hysteresis, that principle I have to apply. So, in the Canny edge detection technique, I have two thresholds, one is the low threshold, another one is the high threshold. And corresponding to this I will be getting two images, one is the g_{NL} and another one is the g_{NH}.

So, I will be getting the weak edge pixels and the strong edge pixels I will be getting. And after this I have to join the edge pixels, that is called the edge linking to get the boundary, to get the continuous edges. So, this is about the Canny edge detection technique and the model-based technique. So, let me stop here today. Thank you.