Digital Image Processing

Prof. P.K. Biswas

Department of Electronics & Electrical Communication Engineering

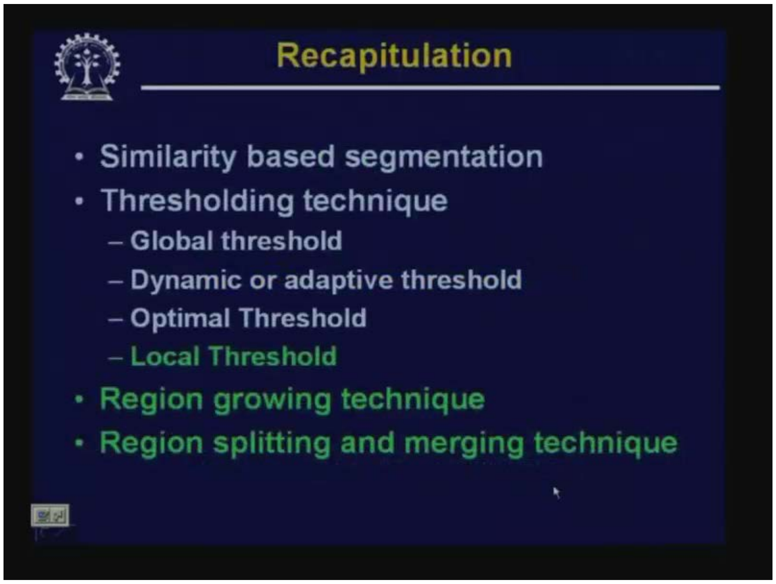
Indian Institute of Technology, Kharagpur

Lecture - 32

Image Segmentation – IV

Hello, welcome to the video lecture series on digital image processing. We are discussing about the image segmentation operations particularly the similarity based image segmentation operations.

(Refer Slide Time: 1:14)



So, we have seen that in similarity based image segmentation operation, there are mainly 3 approaches. One of them is the thresholding based technique where you can go for either global thresholding or dynamic or adaptive thresholding or optimal thresholding or local thresholding.
So, in our last class, we have discussed about the global thresholding operation, we have also discussed about the dynamic or adaptive thresholding operation and we have also discussed about the optimal thresholding operation and we have seen that in case of global thresholding, a threshold value is selected where the threshold value depends only on the pixel intensities in the image.

Whereas, in case of dynamic or adaptive thresholding, it not only depends upon the pixel values or the intensity values or the pixels in the image, it also depends upon the position of the pixel in the image. So, the threshold for different pixels in the image will be different. In case of optimal thresholding, we have tried to find out threshold by assuming that the histogram of the image is a representative of the probability density function of the pixel intensity values.

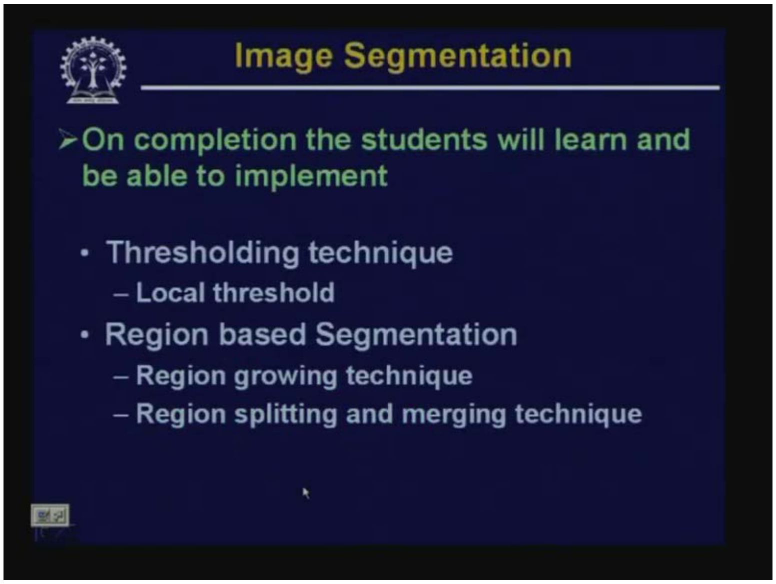
So there, if you have a bimodal histogram, the bimodal histogram is considered as a combination of 2 probability density functions and from the probability density functions, we have tried to estimate that what is the error incurred by performing the threshold operation when an pixel is decided to belong to an object or the pixel is decided to belong to the background.

So, because of the probability distribution function of different intensity values, it is possible that a pixel which actually belongs to the background may be decided to belong to an object or a pixel which actually belongs to an object after thresholding; it may be classified to belong to a background. Now, because of this, there is an amount of error which is incorporated by this thresholding operation.

So, in case of optimal threshold, what we have done is we have tried to estimate that how much is the error incorporated if we choose a particular threshold. Then, you choose that value of the threshold where by which your average error will be minimized. There is another kind of thresholding operation which is the local thresholding operation that we will be discussing today and we have said that local thresholding operation takes care of the neighborhood property or the pixel intensity values in the neighborhood of a particular location (x, y).

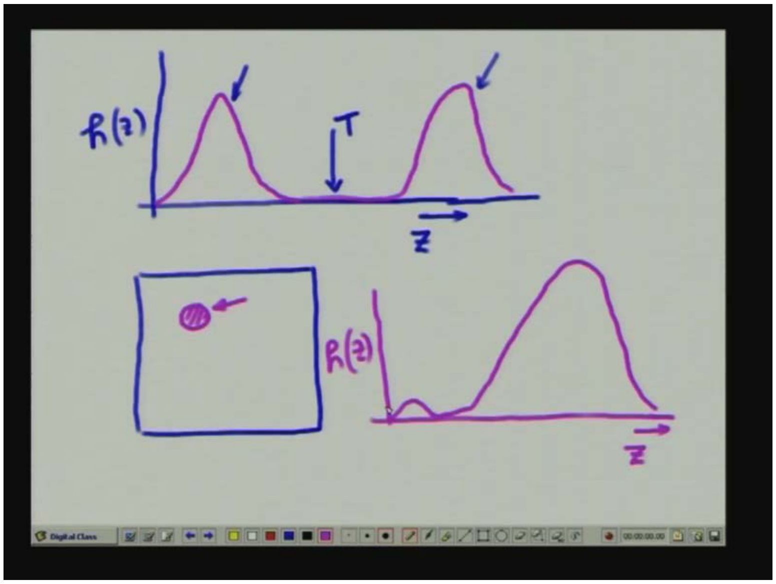
We will also discuss about the other 2 operations, other 2 segmentation, similarity based segmentation operations that is region growing technique and region splitting and merging techniques.

(Refer Slide Time: 4:44)



So, today's discussion will be concentrated on local threshold operations where we will consider in addition to the pixel value, the intensity value, its location; we will also consider the local neighborhood property and the other 2 similarity based segmentation techniques that is region growing technique and region splitting and merging technique.

(Refer Slide Time: 5:50)



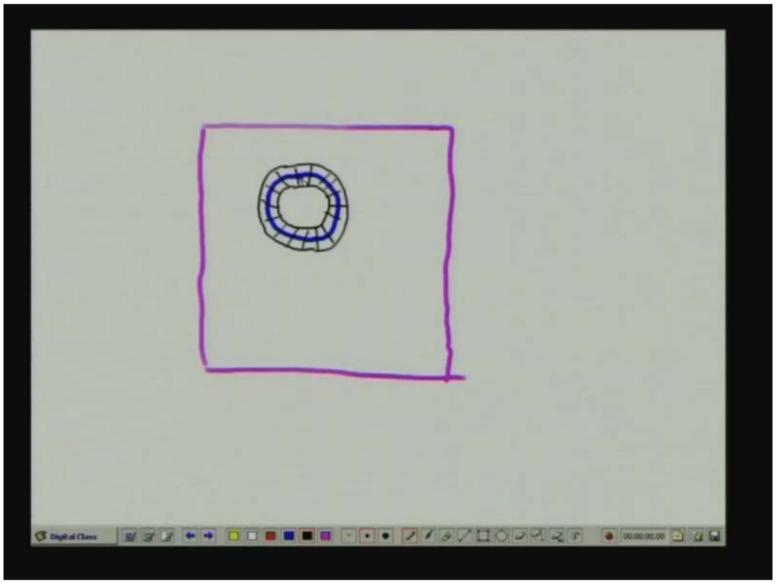
So, first of all let us concentrate on the local thresholding operation. It is now clear that selection of a good threshold value is very simple if the histogram of the particular image is a bimodal histogram where the modes are tall, they are narrow and separated by a div value and in addition, the modes are symmetric. That means if we have a histogram like this; so on this side, we put the pixel intensity values and this side, we put the histogram.

So, if a histogram is of this form, then we can very easily choose a threshold within this valley region. These are the 2 histogram modes or 2 histogram peaks which are separated widely by a value by a valley and within this valley region, we can choose a threshold and by using this threshold, we can segment the image property. But what happens in most of the cases is that the histogram is not so clear. It is not so clearly bimodal and this threshold selection also becomes easy if the histogram is symmetric. That means the area occupied by the object and the area occupied by the background pixels; they are more or less same.

The problem that occurs that if I have an image like this; so I have an image and within this image, very small numbers of pixels actually belong to the object and a large number of pixels belongs to the background and when I have an image like this, the resulting histogram will be something like this. So, this may be the object pixels and the background pixels give rise to a histogram of this form and here you find that the contribution to the histogram by the object pixels is almost negligible because the number of pixels belonging to the object is very small compared to the number of pixels belonging to the background.

So, the bimodal of nature of the histogram is not very visible rather the histogram is dominated by a single mode by the pixels which belong to the background. Now, how to solve this problem?

(Refer Slide Time: 9:01)



So, this problem can be solved if instead of considering all the pixels in the image to produce the histogram, if somehow we can identify the pixels which are either on the boundary or near the boundary between the object and the background in a sense what we are trying to do is that given an image with an object inside, what we are trying to do is we are trying to identify the pixels in a very small trip in a narrow strip around this boundary.

So, if we consider only these pixels around the boundary to form the histogram; the advantage in this case is since we are considering only these pixels near the boundary to form the histogram, the histogram is will be symmetric. That is the area of the pixels within the object region and the area of the pixels and the number of pixels within the background region which are being considered to form the histogram, these 2 numbers of pixels belonging to the object and number of pixels belonging to the background, they will be more or less same, almost same. So, our histogram will be symmetric and it will not be dependent upon the relative size of the object and the background region.

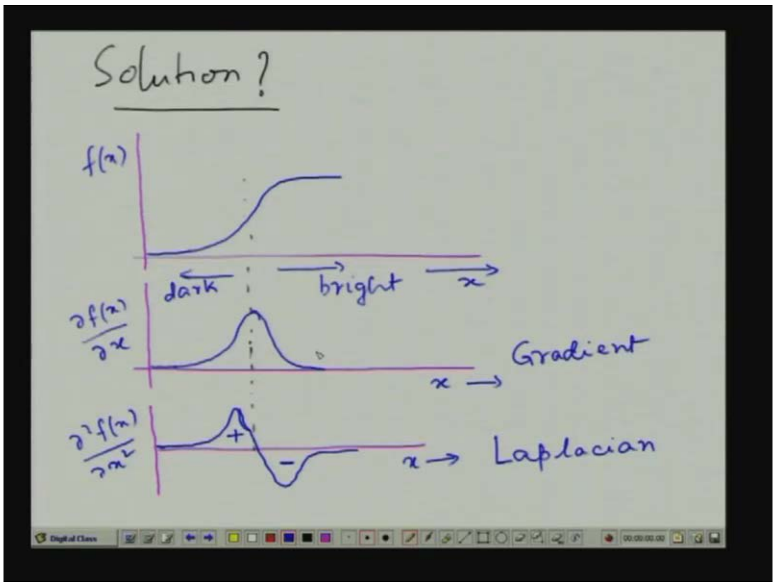
And, the second object is the advantage is the probability of a pixel belonging to the object and the probability of a pixel belonging to the background within this narrow strip, they are almost equal. Because if I consider the entire image, then the probability and in the image, the object region is a very small region; then the property of a pixel belonging to the object is small compared to the probability of the pixel belonging to the background.

Whereas, if I consider the pixels within a narrow strip around the object boundary; in that case, the probability of the pixels belonging to the background and the probability of the pixels belonging to the object they are almost same. So, by considering only those pixels around this narrow strip, I get 2 advantages. One is the pixel belonging the probability of pixel belonging to the background and the probability of the pixel belonging to the object, they are nearly equal and at the same time, the area of the foreground region or the object region and the area of the background region which is used for computation of the histogram that is also nearly same making your histogram a symmetrical histogram. And once I have this kind of histogram, then the thresholding operation is very very simple.

Now, the question is if I simply use this kind of approach; in that case, I have to know that what is the object boundary or what is the boundary between the object region and the background region? But which is not easily obtained because the basic purpose is segmentation, basic purpose of segmentation is that we are trying to find out the boundary between the object and the background.

So, this simple approach as it has been presented that we want to consider the pixels lying on the boundary or the pixels around the boundary; this cannot be used in this simple form because the boundary itself is not known. That is the one that we trying to determine. Then what is the solution?

(Refer Slide Time: 12:50)



So, what is the solution? How do we solve this particular problem? Now, the solution is that if we use the image gradient and the Laplacian image Laplacian, we know that if I have a region something like this, so I plot the variation of intensity values; so this is the pattern of intensity values in an image. So obviously, we are putting it in one dimension, the 2 dimension is now mapped to 1 dimension, so this is my pixel location say x and this is say f(x).

So, this is the variation of intensity along the x direction. If I take the gradient of this and as you know that the gradient is first order derivative operation, so if I compute the gradient of this, the gradient will appear something like this. So again, this is my x direction and on this side, what I am putting is del f(x) del x, this the gradient and also if I take the Laplacian which you know is the second order derivative operator, the Laplacian will appear in this form.

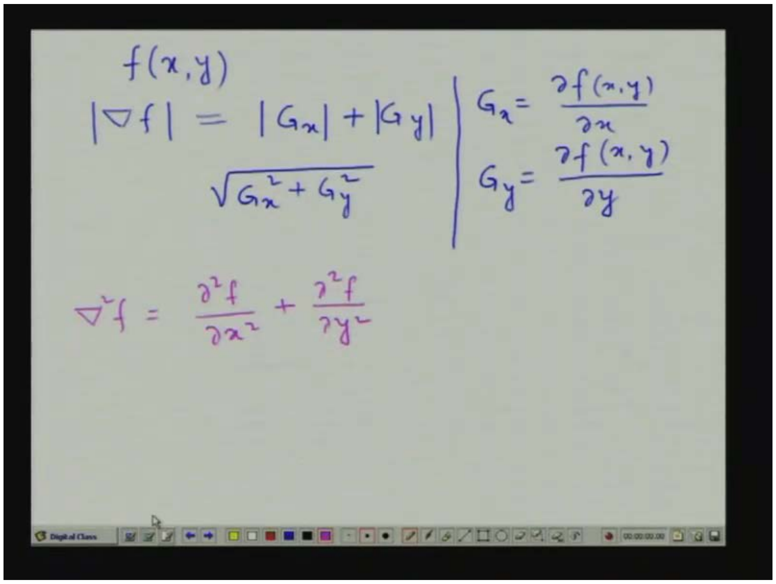
So, this is the second order derivative. Again on this direction, we are putting x; on this direction, we are putting del ^{2} f(x) by del x ^{2}. So, this is f(x), this is gradient and this is Laplacian. So, we have seen earlier that an estimate of the edge points can be obtained from the gradient operator and from the Laplacian operator and we have discussed earlier that the Laplacian operator is affected to a large extent by the presence of noise.

So, the output of the Laplacian operator is not directly used for edge detection purpose but it is used to provide secondary information. So, what we do is you do the gradient operator output to determine the position of the edge points and the output of the Laplacian operator is used to determine whether a point is lying on the darker side of the edge point or it is lying on the brighter side of the edge point.

So, as has been shown here that coming to this intensity distribution, you will find that this is the bright side and this is the dark side and if I compare this Laplacian, you will find that on the bright side of the edge, the Laplacian becomes negative whereas on the dark side of the edge, the Laplacian becomes positive. So, by making use of this information, we can say that whether a point is lying on the dark side of the edge or it is lying on the bright side of the edge.

So, our approach is though we have said that we want to consider only those pixels for generation of the histogram which are lying either on the boundary either on the edge between the object and the background; so, that information can be obtained by using from the output of the gradient because for all the pixels which are lying on the boundary or near the boundary, the gradient magnitude will be quite high and then to decide that out of these points which point lies on the dark side and which point lies on the bright side, we can make use of the Laplacian output where the Laplacian will be negative if a point is lying on the bright side of the edge and the Laplacian will be negative if the point lies in the dark and the Laplacian is positive if the point lies on the dark side of the edge.

(Refer Slide Time: 17:49)

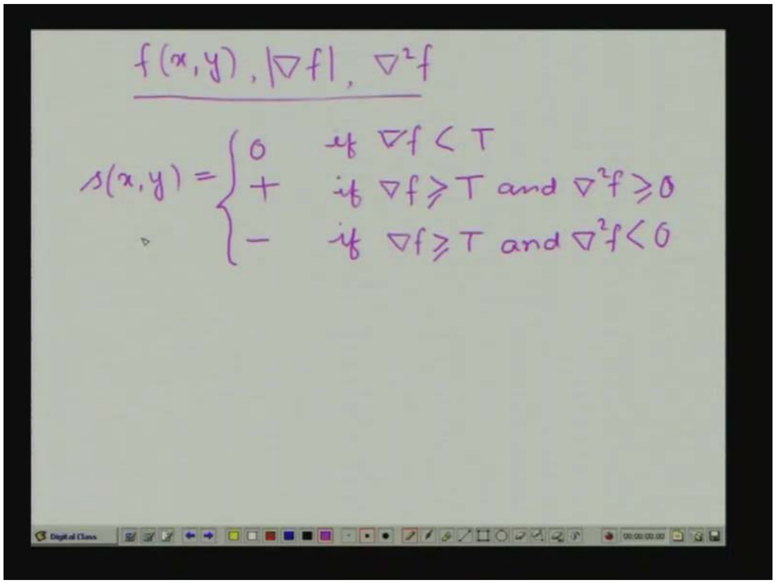


And, we you seen earlier that in case of an image where the image is modeled as a 2 dimensional function f(x, y), the gradient of this image that is grad f, magnitude of this is given by magnitude of G_x plus magnitude of G_y or square root of G_x square plus G_y square where this G_x is nothing but partial derivative of f(x, y) with respect to x and G_y is nothing but partial derivative of f(x, y) with respect to y.

So, G_x is del f(x, y) del x and G_y is del f(x, y) del y and similarly, the Laplacian of this image that is del square f is given by del ^{2} f by del x ^{2} plus del ^{2} f by del y ^{2} and we have seen earlier that to implement this operations in case of digital image, we can have different types of operators differential operators. One of the operator can compute this grad f and the other operator that is Laplacian operator can compute the Laplacian of the given image f(x, y). So here, what we are trying to do is we are trying to estimate whether a point is lying on the edge or the point is within a small region near the edge and then whether the point is lying on the dark side of the edge or it is lying on the bright side of the edge.

So, if I assume that we have an image where we have dark object against a bright background; in that case, for the object pixels, the Laplacian near the edge will be positive and for the background pixel, the Laplacian near the edge will be negative.

(Refer Slide Time: 20:21)



So, simply by making use of this property, what we can do is we can create from f(x, y), then grad of f - gradient of f magnitude of this and del square f from these 3, I can create an image which is say s (x, y) and we will put s (x, y) is equal to 0 if gradient of f is less than some threshold t where it indicates that if the gradient as we have said that on the edge points or the points near the edge, the gradient value will be high.

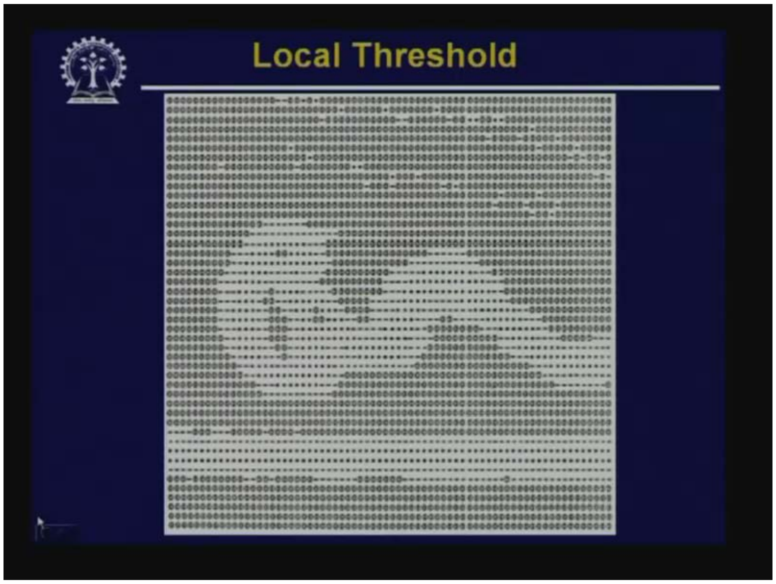
So, if the gradient value is less than some threshold T, we assume that this point does not belong to edge point does not belong to an edge or this point is not even within a region near the edge. So, for such points, we are making s (x, y) is equal to 0 and we will put s (x, y) is equal to positive if gradient of f is greater than or equal to T indicating that this is an edge point or this is a point near the edge and at the same time, if del square f is greater than or equal to 0 which indicates that this point is on the dark side of the edge. That means in this particular case, since we are assuming that we have dark objects against a background; so this is a point on the object side or it is a object point near the object and edge boundary and we will put s (x, y) is equal to negative if it is an edge point or a point near the edge for which again del of f will be greater than or equal to T and the Laplacian that is del square f will be less than 0.

So, what we are doing is we are creating an image s (x, y) which will have values either 0 or positive or negative. Now, for implementation, what we can do is these 3 symbols – 0, positive or negative can actually be represented by 3 distinct intensity values.

So, for example; 0 may be represented by 0, positive may be represented by an intensity value say 128 and negative may be represented by an intensity value say 255. So, 3 distinct intensity values will represent these 3 different symbols – 0, positive and negative and then what we have to do is we have to process this intermediate image s (x, y) to find out the object boundaries of the object regions.

So, here you find that in this representation if s (x, y) is equal to 0 that represents the point does not belong to the boundary, boundary between object and the background if it is positive, then the object belongs to then the pixel belongs to the object region. If it is negative, then the pixel belongs to the background region.

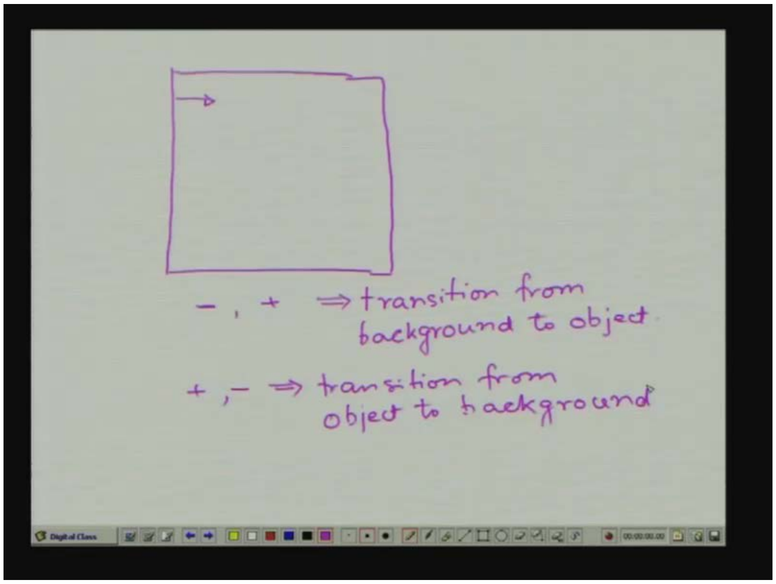
(Refer Slide Time: 24:10)



So, by using this kind of processing, an intermediate image that we can get will be something like this. So, here you find that we have an image which contains one of this 3 symbols either 0, positive or negative and here, what we have done is this was an object, a dark object against bright background may be some handwritten characters with an underline and this information can be processed to find out, this intermediate image can be processed to find out the object region and the background region.

So, once I get an image of this form, you will find that if I scan the image either along a horizontal direction or along a vertical direction, then I am going to get a pattern of these 3 symbols. Now, what will be the nature of this pattern?

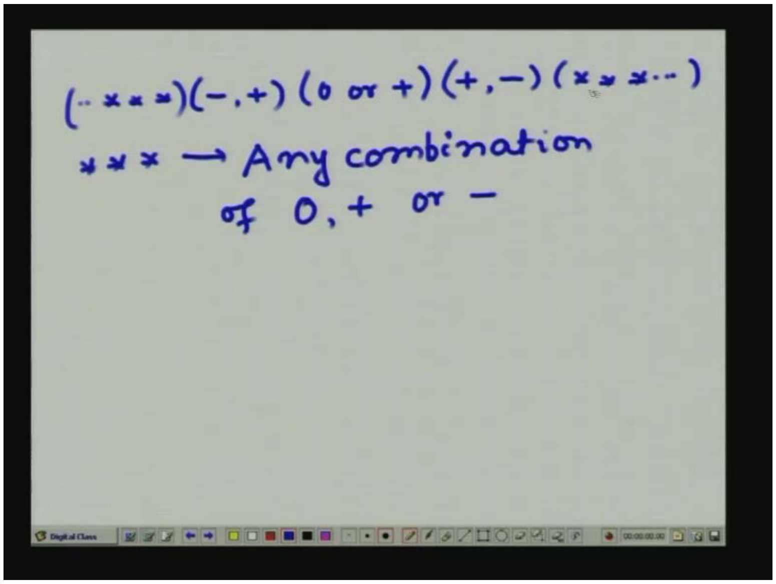
(Refer Slide Time: 25:16)



Say for example, whenever there is an edge, say I have this image, this intermediate image and I want to scan the image along a horizontal line from left to right. Now, while scanning this, since I have assumed that I have dark objects against a bright background; so, whenever there is a transition from the background region to the object region, then I will get a situation something like this. I will get a point having a negative level followed by a point having a positive level. So, a negative followed by a positive, this indicates that I have a transition from background to object.

Similarly, when I am scanning, I am moving from object to the background region; then the combination of these 2 symbols will be just opposite. So here, because I am moving from object region which is dark to the background region which is bright; so the combination of the symbols that I will get is a positive followed by a negative. So, whenever I get this kind of transition that is from a positive to a negative, this indicates that I have a transition from object to background.

(Refer Slide Time: 27:26)

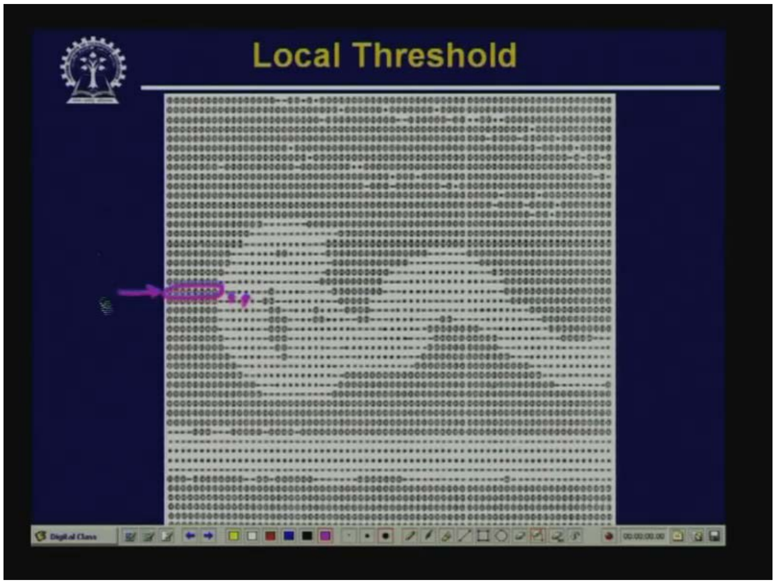


So, by making use of this observation, if I scan a particular horizontal line or a vertical line; then I get a sequence of symbols where the sequence of symbols will be something like this. I will put this as say star, star, star followed by a negative, followed by a positive and then I will have a 0 or positive followed by positive followed by negative and then again a number of stars.

So, if this intermediate image I check either along a horizontal line or along a vertical line and if that particular scan line contains a part of the object; in that case, my scan pattern will be something like this where this star star, this indicates any combination of 0, positive or negative.

So here, you will find that firstly, I can get any combination of 0, positive or negative and then whenever I have a transition from the background region to the object region, I will have a negative followed by a positive and then within the object region, I can have either 0 or positive symbols. Then, when I am moving from the object region to the background region, I can have a transition from positive to negative and then again on the rest part of this scan line, I can have any combination of 0, positive or negative.

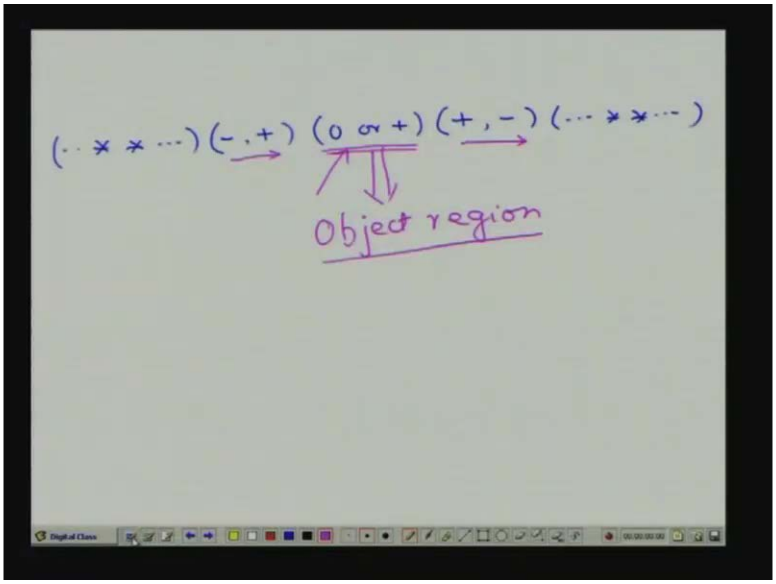
(Refer Slide Time: 29:39)



And you find that this is what is actually represented in this particular image. When I move along any scan line, say for example, I am moving along this particular scan line see if I move along this particular scan line, you will find that initially, I have all 0s, then I have negative symbol followed by I have positive symbol; then within this, it is either 0 or positive, then again I will have a transition from positive to negative, then again I will have a number of 0's and this is how it continues.

So, by making use of this particular pattern, I can identify that on this particular scan line which part is which portion of this scan line belongs to the object and which portion of this scan line belongs to the background.

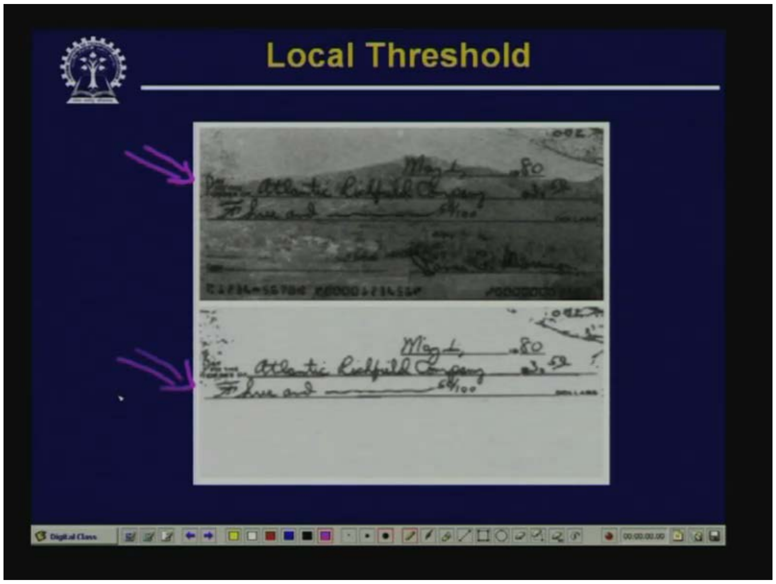
(Refer Slide Time: 30:53)



So, the kind of scan lines or symbols on the scan lines that we have obtained is like this. First, I have any combination of positive, 0 or negative; then I have negative, positive; then I have either 0 or positive; then I have positive followed by negative and then again I can have any combination of 0, positive or negative and here you find that this inner parenthesis, this transition from 0 to positive or from positive to 0, this indicates the occurrence of edge points and this inner parenthesis where I have only 0 or positive symbols, these actually indicates the object region.

So, for segmentation purpose, what I can do is when I scan this intermediate image s (x, y) either along a horizontal line or along a vertical line; then only these part of the scan line which is represented by this inner parenthesis, all those points I will make equal to 1 and rest of the points on this scan line, I will make equal to 0 and that gives me a segmented output where in this output image all the scan lines or the part of the object is represented by a pixel value equal to 1 and all the pixel all the background pixels background regions are represented by a pixel value equal to 0.

(Refer Slide Time: 32:59)



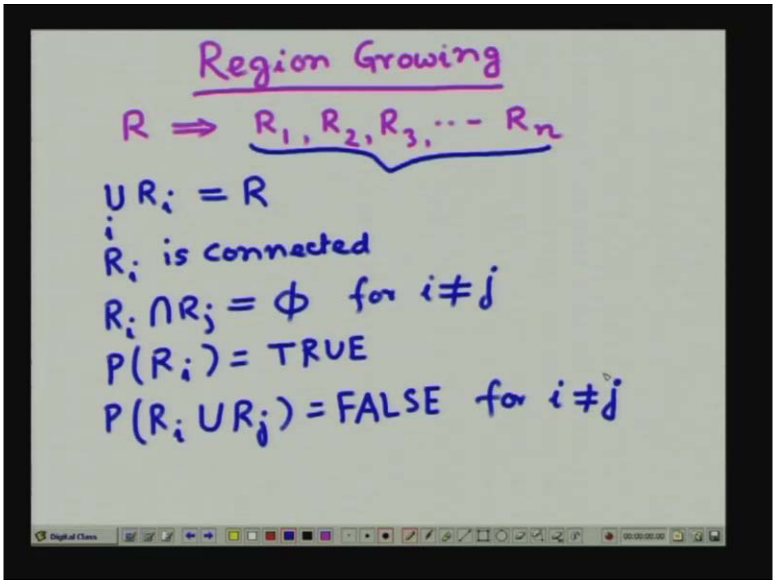
So, if I apply this technique on an image, you will find what kind of result that we get. You find that in this particular case, this is an image, the top part of it is the scanned image of a bank cheque and here you will find that the signatures and the other figures, they are appearing in a background and it is not very easy to distinguish which is the object or which is the signature or figure part and which is really the background part.

And, by making use of this kind of processing and filling all the object regions either with 0 or 1; we can clearly segment out the signature part and the figure part and here you will find that this kind of output possibly we cannot get by making use of any of the global thresholding approach.
But here, by using this local thresholding and we call it local thresholding because to find out this threshold what we have made use of is the gradient of the image and the Laplacian of the image. And, the gradient and Laplacian, these are local properties local to a particular pixel location.
So, the kind of thresholding which is inbuilt in this kind of segmentation operation that is what we call as local thresholding. So, with this we have discussed about the different kind of thresholding operations. In our earlier class, we have discussed about global thresholding, we have discussed about the dynamic or adaptive thresholding, we have discussed about the optimal thresholding and now what we have discussed about is the local thresholding where this local thresholding operation makes use of the image gradient and image Laplacian and as we said that this gradient and Laplacian, these are local properties to a particular pixel location.

So, the kind of thresholding which is embedded in this application is nothing but a local thresholding operation. Though this segmentation operation is obtained by scanning the intermediate image that is generated, there is no direct thresholding operation involved in it. But the kind of operation that is embedded in this approach is nothing but what we call as local thresholding operation. Now, let us go to the other approaches of segmentation.

We have said there are 2 other approaches of similarity based segmentation operations; one of them is region growing segmentation, the other one is called splitting and merging segmentation operation. So, first let us talk about the region growing operation.

(Refer Slide Time: 35:51)



Now, what is this region growing segmentation? It is like this that suppose, we consider that all the pixels belonging to the image as a set of pixels say R and by this region growing operation or the segmentation operation, what it does is it partitions this set of pixels R into a number of sub regions say R_1 R_2 R_3 like this upto say R_n.

So, what segmentation is doing is segmentation operation is actually partitioning this set of pixels R which actually represent the entire image into a number of sub images or partitions that is n number of partitions R_1 to R_n. Now, when it is doing this kind of partitioning that is when I partition my original set R into n number of such partitions R_1 to R_n, these partitioning should follow certain property.
The properties are if I take the union of all these regions R_i union over i, this should give me the original image R. That means none of the pixels in the image should be left out, it is not that some pixel is not part of any of the partitions. So, every pixel in the image should be a part of one of the partitions. The second property is the region R_i should be connected and we have defined earlier that what do we really mean by a connected region.

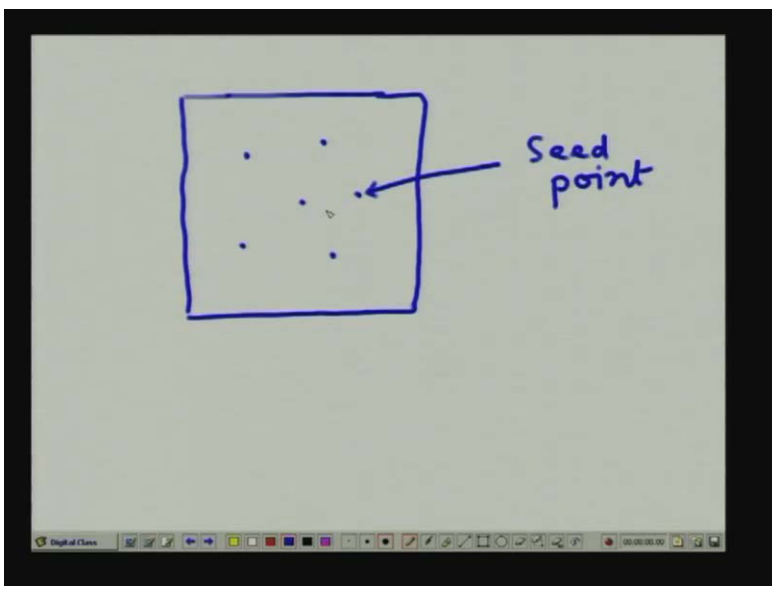
We have said that given a region, the region will be called connected if I take any 2 points in the region; then I should be able to find out a path between these 2 points considering only the points which are already belonging to this region R. So, if every pair of points in this region R_i are connected, then we say that this region R_i is connected. So, the second property that this segmentation or partitioning must follow is that this partitions we get, n number of partitions, every partition R_i should be connected.

The third property that must be followed is R_i intersection R_j that should be equal to null for i not equal to j. That means if I take 2 partitions say R_1 and R_2, this R_1 and R_2 should be disjoined. That means there should not be any common pixels, any common points in these 2 partitions - R_1 and R_2. Then, if I define a predicate say P over a region R_i that should be true where this P is a logical predicate defined over the points in set R_i in this partition R_i. So, for a single partition R_i, this logical predicate P should be true and the last property that must be followed is predicate over R_i union R_j that must be equal to false.

So, what does it mean? False for i not equal to j; so, this actually means that if I define a predicate for the pixels of the points belonging to a particular region, then the predicate must be true for all the points belonging to that particular region and if I take points belonging to 2 different regions R_i and R_j, then the predicate over this combined set R_i union R_j must be equal to false. So, this is what says the similarity. That means all the points belonging to a particular region must be similar and the points belonging to 2 different regions are dissimilar.

So, what does this region growing actually mean? The region growing as the name implies that it is a procedure which groups the pixels or sub regions into a larger region based on some predefined criteria and in our case this, predefined is the defined predicate. So, we start from a single point and try to find out what are the other points that can be grouped into the same group which follows the same criteria or for which the predicate value is or for all of which the predicate is true. So, this region growing operation works like this.

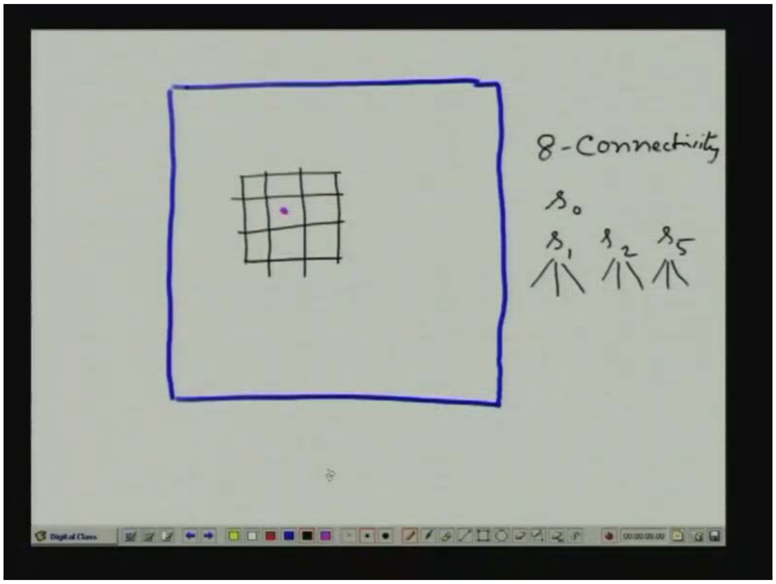
(Refer Slide Time: 42:13)



I have image and in this image, I may select a set of points. So somehow, I select a set of points like this and each of these points, I call as a seed point and then what region growing operation tries to do is it tries to grow the region starting from the seed point by incorporating all the points which are similar to the seed point.

Now, the similarity measure can be of different types. For example, we can say that 2 points are similar if their intensity values are very close and the points are dissimilar if their intensity values are widely different and one of the conditions that we have said that the points must be connected.

(Refer Slide Time: 43:14)



That means coming to this image again, say I have this big image and for region growing, what I have to do is I have to choose a seed point and our region growing operation will start from the seed point. So, for this purpose, what I will to do is I can define I can have a 3 by 3 neighborhood around this seed point and since one of the property that this partitions have to follow; so what I am doing is I am choosing this 3 by 3 neighborhood around the seed point and since one of the property that this partitions have to follow is that every region or every partition has to be connected. That means when I start to grow the region starting from the seed point, then all the points which I will include in the same group or in the same partition, these points have to be connected. That means I have to start growing this region from the points which are connected to the seed point.

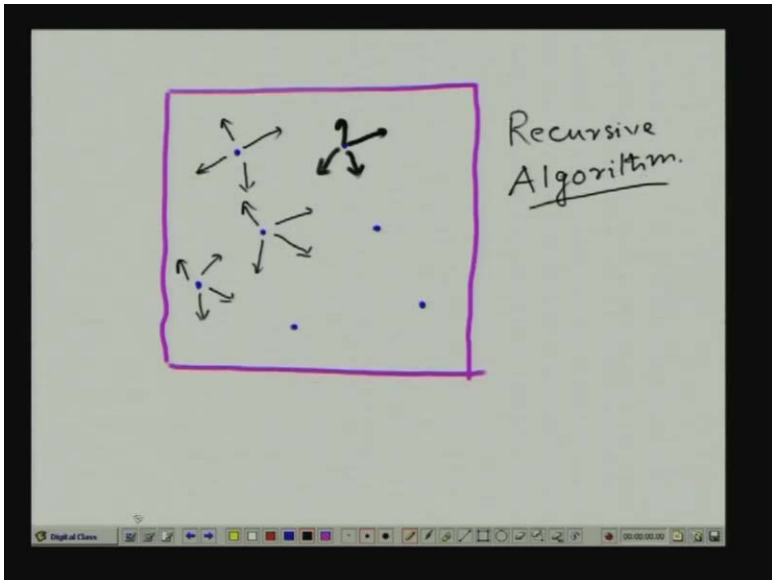
So here, if I use the 8 connectivity, concept of 8 connectivity; in that case, the point which are to be grouped or in the same group as this seed point, they have to be one of they must belong to this 3 by 3 neighborhood of this particular seed point. So effectively, what I am doing is once I choose a seed point, I check the points in its 3 by 3 neighborhood and all the points which I find are similar to this seed points, those points are put in the same group and then again I start growing the region from all these new points which are put in the same group.

So effectively, what I am doing is if I call this seed point which is put which has been selected as seed point s_0. Now, from its neighborhood, I may find that the other points which can be put in the same group as point as this initial seed point s_0 are say s_1 s_2 and say s_5. Next time I start growing the region from s_1 itself. I find within the neighborhood of s_1, within this 3 by 3 neighborhood of s_1 following the same 8 connectivity; what are the points which are similar to s_1

or what are the points which are similar to the seed point? And, this similarity can be based on the intensity difference.

If the intensity difference is small, I say that they are similar; if the intensity difference is high, I say that they are not similar. So, by this, again I start growing the region from s_1, I start growing the region from s_2, I start growing from region from s_5 and so on and this process will stop when no more new point can be included in the same row.

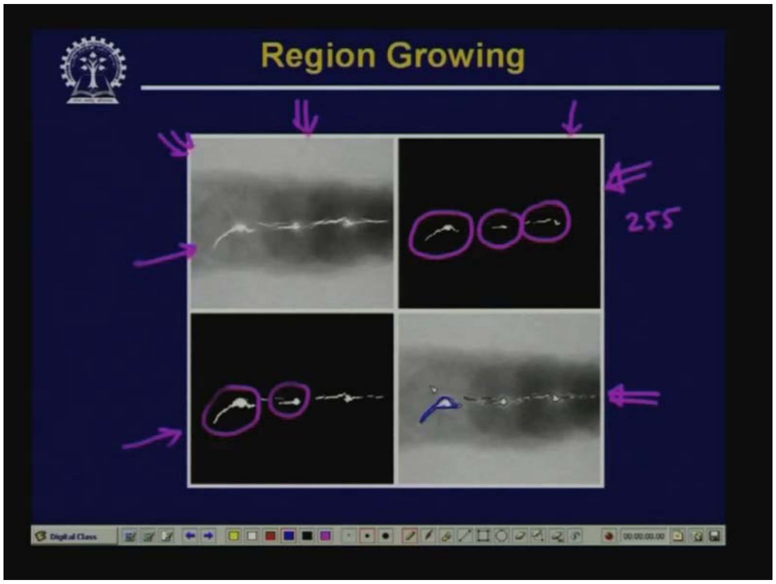
(Refer Slide Time: 46:28)



So effectively, what we are doing is we are selecting a number of seed points in the image following some criteria. So, we have selected a number of seed points. So, this seed point selection is application dependent and once you select the seed points; then from the seed points, we start growing the region in different directions by incorporating more and more points which are connected as well as similar. And at the end, what we have is a number of regions which are grown around these seed points.

So, this is what is the basic region growing operation and you will find that this basic region growing operation can be very easily implemented by using some recursive algorithm. Now, let us see that what kind of output or result we can get by using this region growing segmentation operation?

(Refer Slide Time: 47:52)



So, here is an example. This is the x ray image taken from a weld and you will find that in case of this x ray image, there may be some cracks within the welded region or there may be some faults within the welded region and these faults can be captured by using an x ray image. So, this top left one, this is the x ray image of a welded part and the nature of the problem says that whenever there is a fault, then the faulty regions in the x ray image are going to have very high intensity values.

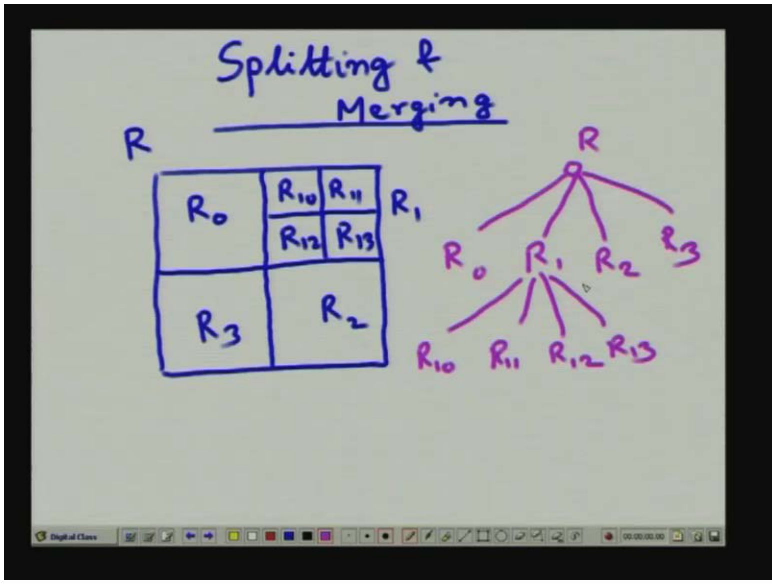
So here, on the left hand side, it is first a simple segmentation operation, the thresholding best segmentation operation where these are the pixel values, these are the regions having pixel values near an intensity value of 255 that is the maximum intensity value and as we said that these faults usually appear as higher intensity values in the x ray image.

Then, what you do is the seed points are actually selected as all the points in this thresholded image having a value of 255 after the thresholding operation and then you start the region growing operation around each of these seed points. So, if I grow the region around each of the seed points; now when you go for this region growing operation, the region growing operation has to be done on this original image not on the thresholded image. This thresholding operation is simply done to select the seed points.

Once you get the seed points, come to the corresponding seed point location in your original x ray image and grow the region starting from those seed locations within the original x ray image and this one shows the grown regions and now you will find that if I super impose the boundary of these grown regions, each of them are the grown region; so, if I super impose the boundary of this grown region on this original x ray image, this super position output is shown on the bottom right image.

Here, you will find that these are actually the boundary regions boundaries which are super imposed on the original image. So, you will find that your segmentation operation in this particular case is quite satisfactory. So, by using this similarity measure and incorporating them along with the region growing operation, we can have quite satisfactory segmentation operation. So, the next type of segmentation that we said that we will discuss about is splitting and merging operation, splitting and merging.

(Refer Slide Time: 50:52)

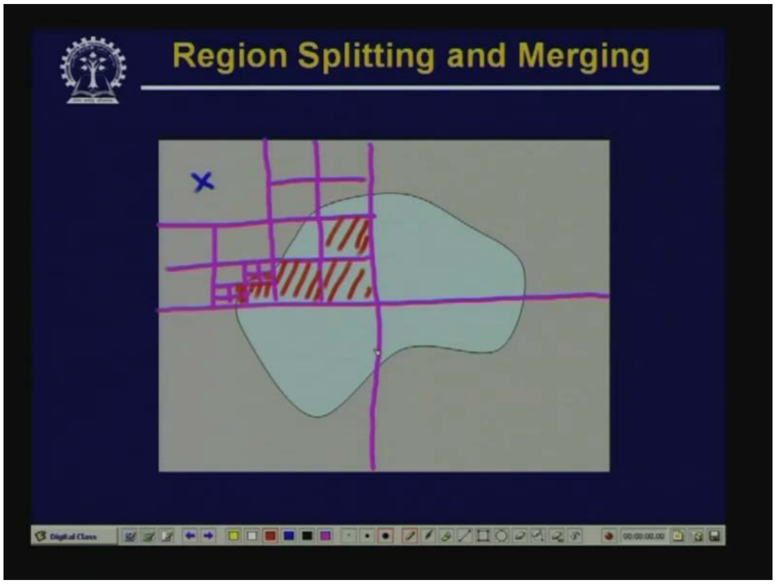


Here again, what we are trying to do is we are making as trying to form a segment of all the pixels which are similar in intensity values or similar in some sense. Our approach in this particular case will be that if I have an image say R, first you try to find out whether this entire image region is similar or not or whether the intensity values are similar. If they are not similar, then you break this image into quadrants. So, just make 4 partitions of this image. Then you check each and every partition in this image. If they are similar, if all the pixels within a partition are similar; you leave it as it is. If it is not similar, then again you partition that particular region.

So initially, suppose, this was region R_0, this was region say R_1, this was region say R_2, this was region say R_3; now this R_1 is not uniform, so I partition that again making it R_{10} R_{11} R_{12} and say R_{13} and you go on doing this partitioning until and unless you come to a partition size which is the smallest size permissible or you come to a situation where the partitions have become uniform, so I cannot partition them anymore. And in the process of doing this, what I am doing is I am having a quadratary representation of the image.

So, in case of quadratary representation, if root node is R, my initial partition gives me 4 nodes - R_0 R_1 R_2 and R_3. Then R_1 I am partitioning again in R_{10} R_{11} R_{12} and R_{13}. Once such partitioning is completed, then what you do is you try to check all the adjacent partitions to see if they are similar. If they are similar, you merge them together to form a bigger segment. So, this is the concept of splitting and merging technique for segmentation. Now, let us see this with the help of an example.

(Refer Slide Time: 53:49)



See, I have an image of this form. When you come to this original image, you will find that here I have this back ground and on this background, I have this object region. This is obviously non uniform. So, what I do is I partition them into 4 quadrants, each of them are non uniform. So, I have to partition them again.

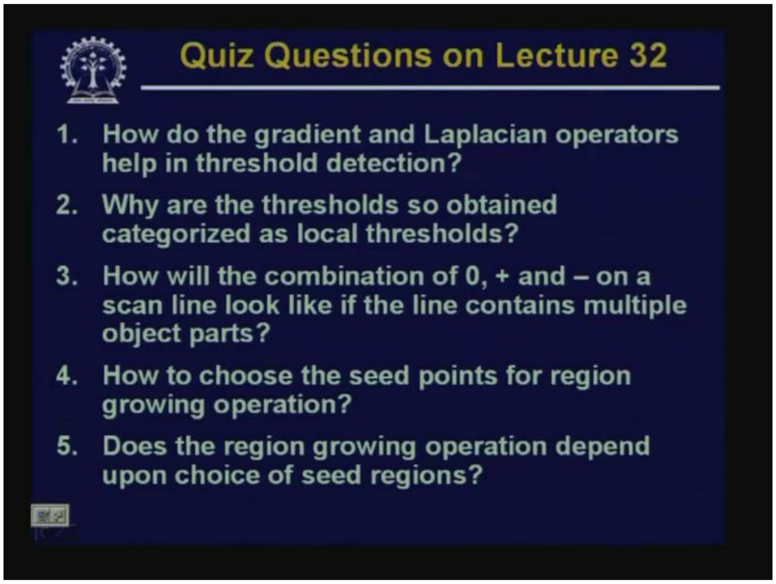
So, let us take one particular partition example of one particular partition; so, I partition them in four again and here you will find that this particular partition is uniform. So, I do not partition it any more. The rest of the partitions, I have to go for sub partitioning like this. Let us take one of them; this is partitioned again, this is partitioned again, this is partitioned again and so on.

Now at the end, when I find that I cannot do any more partitioning again either I have reached a minimum partition size or every partition has become uniform; then I have to look for adjacent partitions which can be combined together to give me a bigger segment. So, that is what I do in this case. Here, you find that this partition, this partition, this partition, this partition and this partition, they can be grouped together. So and then again, this particular group can be combined with this particular partition, it can be combined with this partition, it can be combined with this partition and so on.

So finally, what I get is after splitting after the splitting operation; the entire object, I break into a number of smaller size partitions and then in the merging operation, I try to find out the partitions which can be merged together to give me a bigger segment size. So, by doing this at the end of this splitting and merging operation, different objects can be segmented out from the background.

So, in brief, we have discussed about the different segmentation operations. Initially, it started with discontinuity based segmentation where we have gone for different edge detection operation or line detection operation followed by linking and then we have discussed about the similarity based segmentation. Under similarity based segmentation, we have discussed about various thresholding operations, the region growing operation and lastly, the splitting and merging operations. Now, let us have some quiz questions on today's lecture.

(Refer Slide Time: 56:40)



The first question, how do the gradient and Laplacian operators help in threshold detection? Second question, why are the thresholds so obtained categorized as local thresholds? Third question, how will the combination of 0, positive and negative on a scan line look like if the line contains multiple object parts? Fourth question, how to choose the seed points for region growing operation? And the last question, does the region growing operation depend upon the choice of seed regions or does the result of region growing operation depends upon the choice of seed regions?

Thank you.