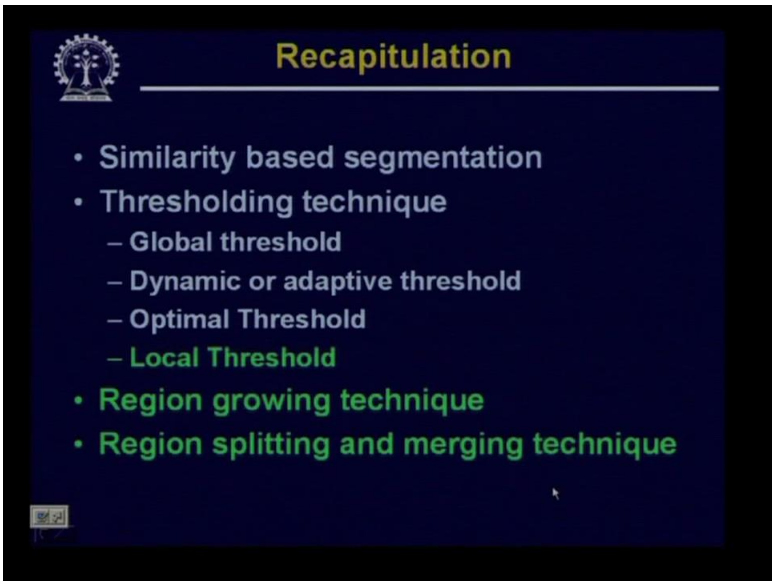
Digital Image Processing.
Professor P. K. Biswas.
Department of Electronics and Electrical Communication Engineering.
Indian Institute of Technology, Kharagpur.
Lecture-59.
Region Splitting and Merging Technique.

(Refer Slide Time: 0:35)



Hello, welcome to the video lecture series on digital image processing. We are discussing about the image segmentation operations, that in similarity based image segmentation operation there are mainly three 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.

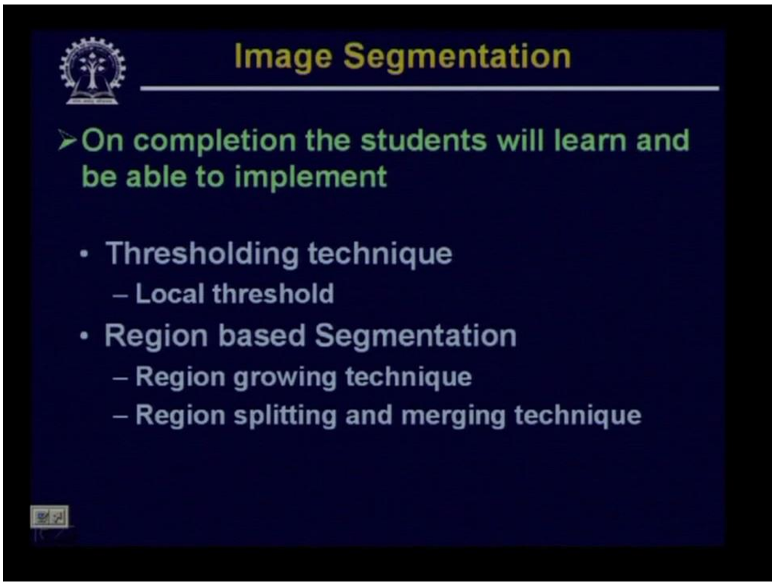
So, the threshold for different pixels in the image will be different. In case of optimal thresholding we have tried to find out threshold by assuming that the histogram of the image is a representative of the probability density function of the pixel intensity values. So, there if you have a bimodal histogram, the bimodal histogram is considered as a combination of two 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 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).

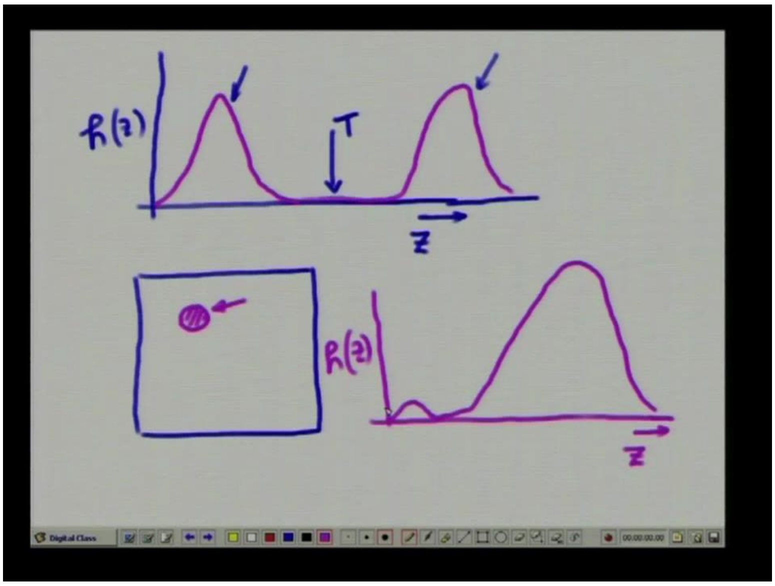
We will also discuss about the other two operations other two segmentation similarity based segmentation operations that is region growing technique and region splitting and merging techniques.

(Refer Slide Time: 4:12)



So, our 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 two similarity based segmentation techniques that is region growing technique and region splitting and merging technique.

(Refer Slide Time: 5:22)



So, first of all let us concentrate the local thresholding operation it is now clear that selection of a good threshold value is a 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 deep value valley and in addition the modes are symmetric. That means if we have a histogram of like this, so, on this side we put the pixel intensity values and on this side we put the histogram.

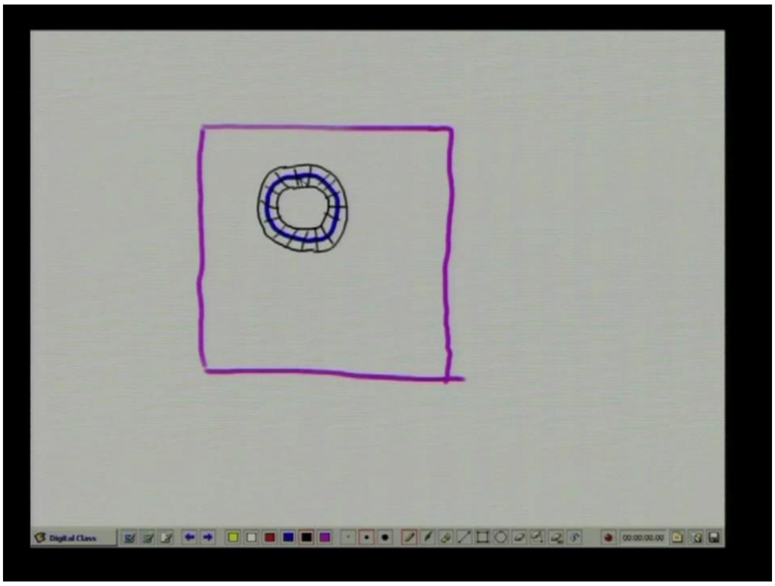
So, if an histogram is of this form then we can very easily choose a threshold within this valley region. These are the two histogram modes or two histogram peaks which are separated widely by value by a valley and within this valley region we can choose a threshold. And by using this threshold we can 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 is 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 number of pixels actually belong to the object and a large number of pixels belong 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 continuation 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 a 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: 8:34)



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. Okay. What we are trying to do is? We are trying to identify the pixels in a very small strip 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 two number of pixels belonging to the object and the number of pixels belonging to the background, they will be more or less same almost same.

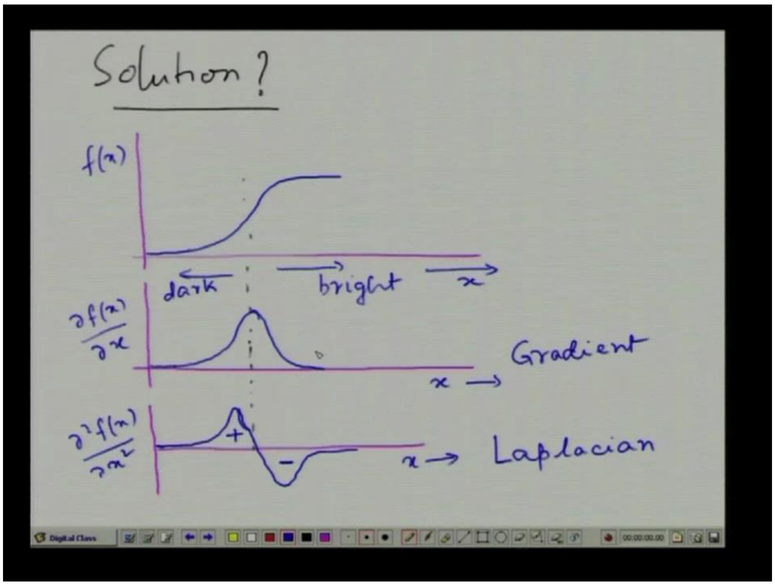
So, our histogram will be symmetric and it will not be dependent upon the relative size of the object and the background. And the second object is the advantage is, the probability of a pixel belonging to the object and the probability of a pixel belonging to the background within this narrow strip they 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 probability of a pixel belonging to 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 pixel is belonging to the object they are almost same.

So, by considering only those pixels around this narrow strip I get two 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 of 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 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 are trying to determine. Then what is the solution? So, what is the solution? How to we solve this particular problem?

(Refer Slide Time: 12:21)



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 two dimension is now mapped to one dimension. So, this is my pixel location say x and this is the f(x).

So, this the variation of intensity along the x direction if I take the gradient of this as we know that the gradient is first order derivative operation. So, if I compute the gradient of this the gradient will appear something like this. Okay. So, again this is my x direction and on this side what I am putting is del f(x) delx it is 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 delx 2. So, this is the 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 operation and we have discussed earlier that the laplacian operator is affected with large extent by the presence of noise.

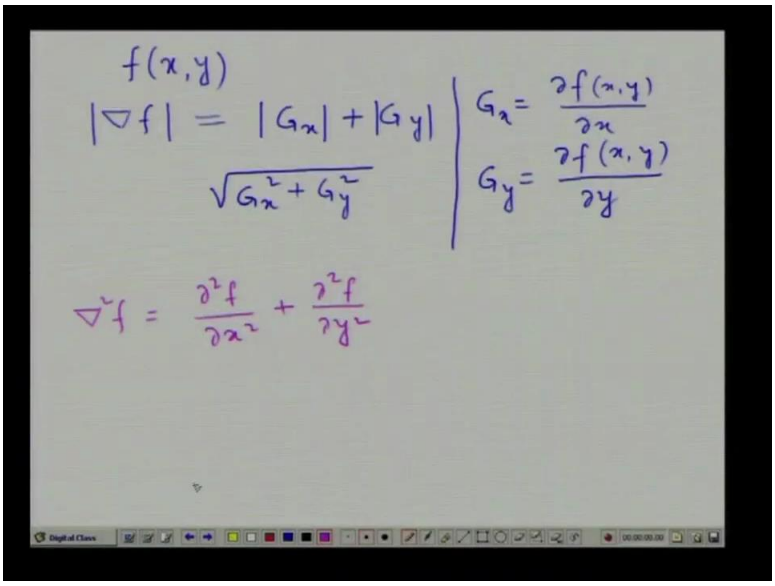
So, the output of the laplacian operator is not directly used for edge detection purpose but it is used to provide secondary information. So, what we do is you do the gradient operator output to determine the position of the edge points and the output of the laplacian operator is used to determine whether 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 find that this is the bright side and this is the dark side and if I compare this laplacian you 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. Okay. 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 points lies on the dark side and which points lies in 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 on the dark, and the laplacian is positive if the point lies on the dark side of the edge.

(Refer Slide Time: 17:15)

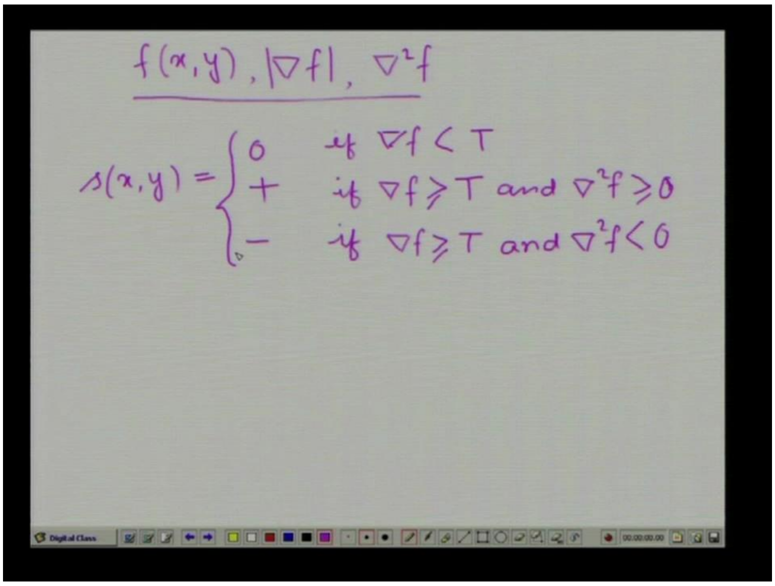


And we have seen earlier that in case an image or if the image is modeled as a two dimensional function f(x, y) the gradient of this image that is grad f magnitude of this is given by magnitude of Gx plus magnitude of Gy or square root of Gx square plus Gy square, where this Gx is nothing but partial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

So, Gx is del f(x, y) by delx and Gy is del f(x, y) by del y and similarly, the laplacian of this image that is del square f is given by del two f by del x two plus deltwo f by del y two. And we have seen earlier that to implement these 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 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: 19:41)



So, simply be 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 three I can create an image which is say s (x, y) and we will put s (x, y) is equal to zero if gradient of f is less than some threshold T, but 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.

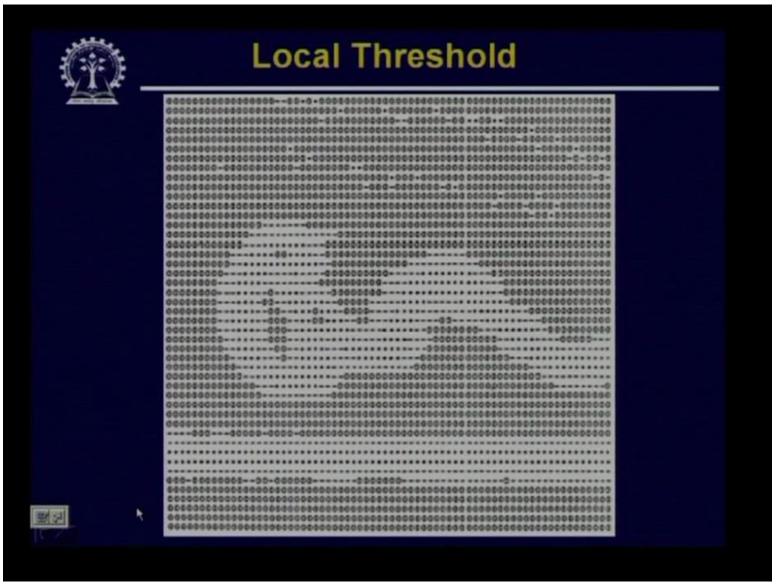
So, if the gradient value is the less than some threshold T we assume that this point does not belong to edge point does not belong to an edge or this point is not even within a region near the edge. So, for such point we are making s (x, y) is equal to zero and we will put s (x, y) is equal to positive if gradient of f is greater than or equal to T indicating that this is an edge point or this is the point near the edge. And at the same time if del square f is greater than or equal to zero 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 so or it is a object point near the object and the 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 zero. So, what we are doing is we are creating an image s (x, y) which will have values either zero or positive or negative.

Now, for implementation what we can do is these three symbols zero, positive or negative can actually be represented be three distinct intensity values. Say, for example, zero may be represented by zero, positive may be represented by an intensity value say 128 and negative may be represented by an intensity value say 255. So, three distinct intensity values will represent these three different symbols zero, 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 zero 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 the then the pixel belongs to the object region, if it is negative then the pixel belongs to the background region. So, by using this kind of processing an intermediate image that you can get will be something like this.

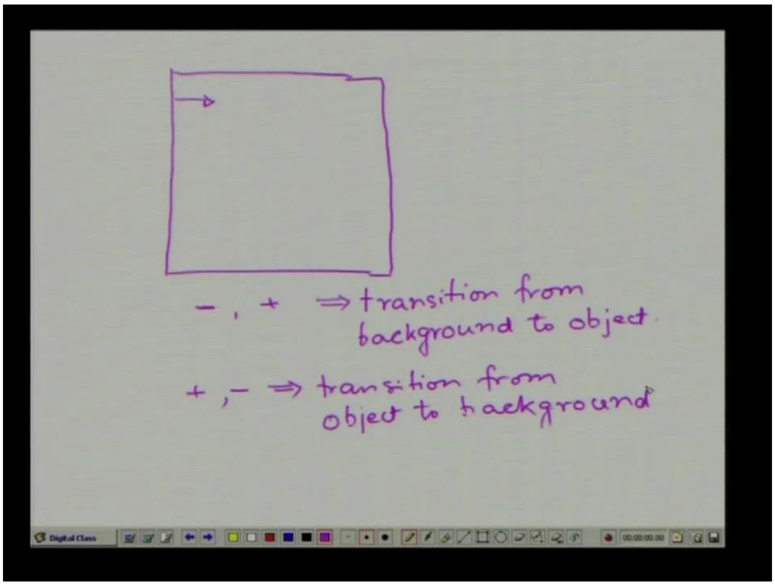
(Refer Slide Time: 23:33)



So, here you find that we have an image which contains one of these three symbols either zero, positive or negative. And here, what we have done is, this was an object 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 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 three symbols. Now, what will be the nature of this pattern? Say, for example, whenever there is an edge, so, I have this image this intermediate image and I want to scan the image from along a horizontal line from left to right.

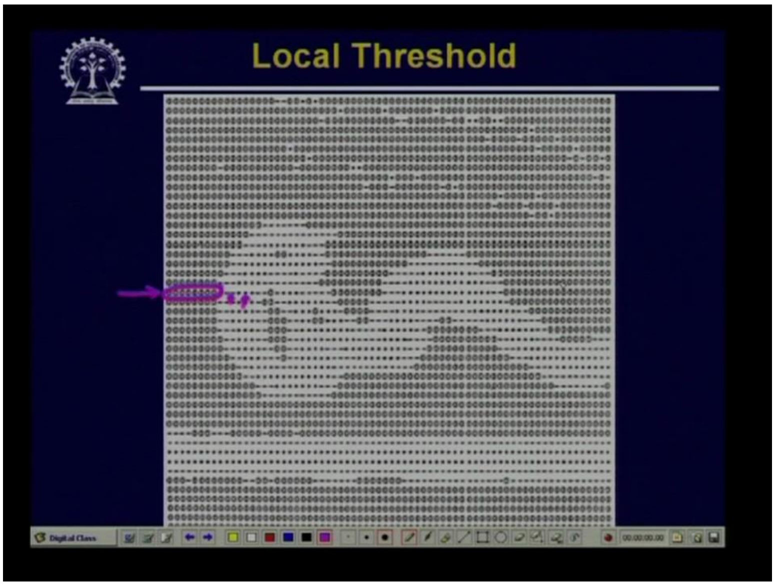
(Refer Slide Time: 24:56)



Now, while scanning this since I have assumed that I have dark objects against the 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 two 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.

(Refer Slide Time: 26:46)



So, whenever I get this kind of transition that is from a positive to a negative this indicates that I have transition from object to background. 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 zero or positive followed by positive followed by negative and again, then again a number of stars.

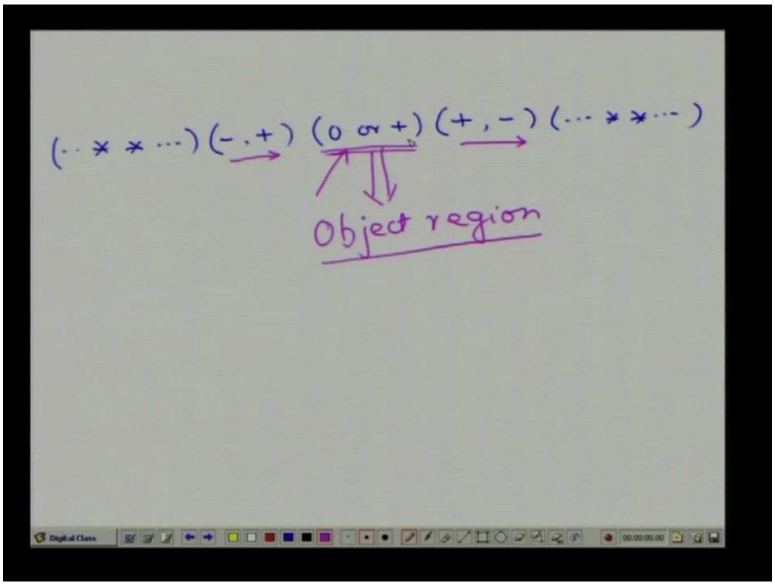
So, if this intermediate image I check either along a horizontal line or along a vertical line, and if that particular scan line contains a part of the object, in that case my scan pattern will be something like this, where this star, star, this indicates any combination of zero, positive or negative. Okay. So, here you find that firstly I can get any combination of zero, 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 zero 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 zero, positive or negative 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. So, if I move along this particular scan line, you will find that initially I have all zeros then I have negative symbol followed by I have positive symbol. Then within this it is either zero or positive, then again I will have a transition from positive to negative, then again I will have a number of zeros and this is how it continues.

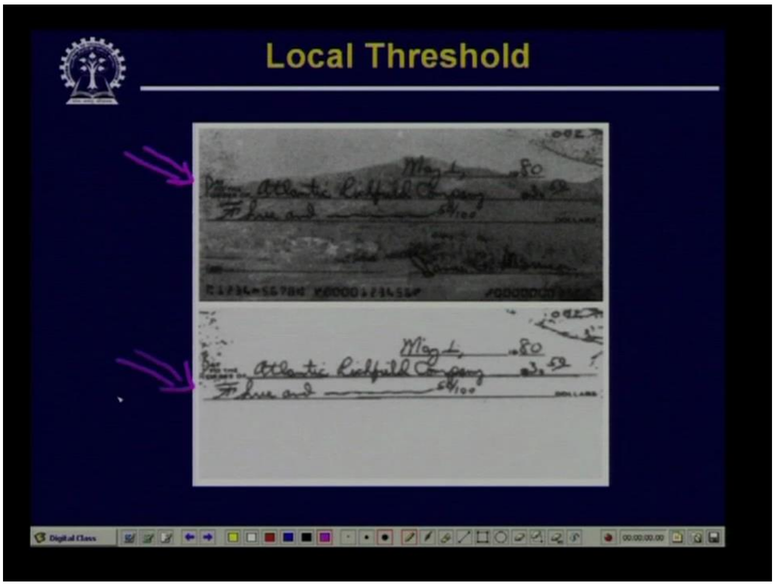
So, by making use of this particular pattern I can identify that on this particular scan line which portion of the scan line belongs to the object and which portion of the scan line belongs to the background. So, the kind of scan lines or a symbol on the scan lines that we have obtained is like this. First I have any combination of positive, zero or negative, then I have negative, positive. Then I have either zero or positive then I have positive followed by negative and then again I can have any combination of zero, positive or negative. And here you find that this inner parenthesis.

(Refer Slide Time: 31:18)



This transition from zero to positive or from positive to zero this indicates the occurrence of edge points. And this inner parenthesis where I have only zero 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 this part of the scan line which is represented by this inner parenthesis all those points I will make equal to one and the rest of the points on this scan line I will make it 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 one and all the background pixels, background regions are represented by a pixel value equal to zero.

(Refer Slide Time: 32:21)



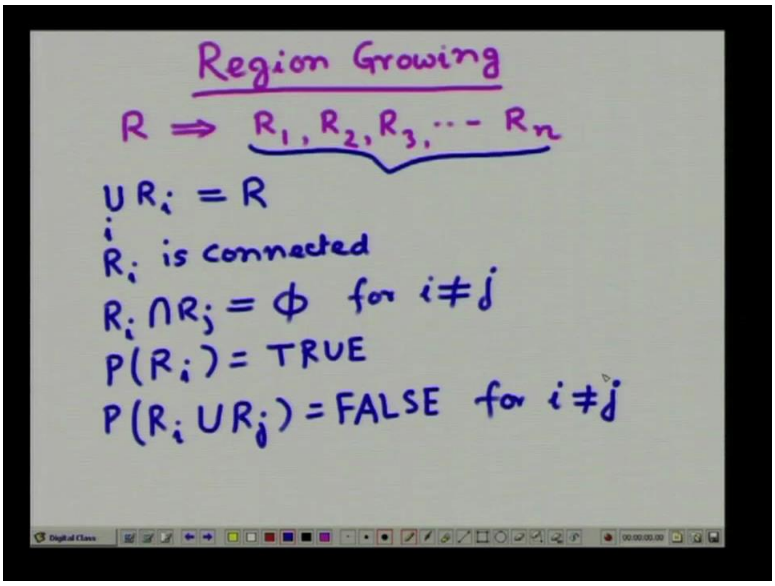
So, if I apply this technique on an image you find what kind of result that we get. You find that in this particular case this is an image, the top part of it, is the scanned image of a bank cheque. And here, you find that the signatures and other figures, they are appearing in a background and it is 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 a use of this kind of processing and filling all the object regions either with zero or one, we can clearly segment out the signature part and the figure part.

And here, you 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 a 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 bout the optimal thresholding and now, what we have discussed about is the local thresholding. But 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 a local thresholding operation. Now, let us go to the other approaches of segmentation, we have said there are two other approaches of similarity based segmentation operations. One of them is region growing segmentation the other one is called splitting and merging segmentation operation.

(Refer Slide Time: 35:12)



So, first let us talk about the region growing operation. 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 pixel say R. Okay 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 R1, R2, R3 like this up to say Rn. So, what segmentation is doing is, segmentation operation is actually partitioning this set of pixels R which actually represents the entire image into a number of sub-images or partitions that is n number of partitions R1 to Rn.

Now, when it is doing this kind of partition that is when I partition my original set R into n number of such partitions R1 to Rn. This partitioning should follow certain property. The properties are if I take the union of all these regions Ri, 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 other partitions.

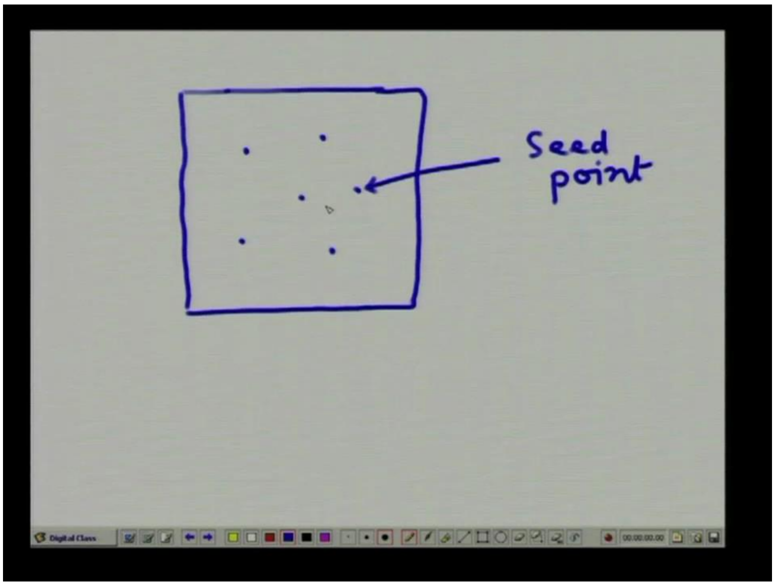
So, every pixel in the image should be a part of one of the partitions. Okay. The second property is the region Ri should be, is connected. And we have defined earlier that what do we really mean by a connected region. We have said that given a region, the region will be called connected if I take any two points in the region then I should be able to find out a path between these two points, considering only the points which are already belonging to this region Ri.

So, if every pair of points in this region Ri are connected, then we say that this region Ri is connected. Okay. So, the second property that this segmentation or partition you must follow is that these partitions we get n number of partitions every partition Ri should be connected. The third property that must be followed is Ri intersection Rj that should be equal to null for i not equal to j. That means if I take two partitions, say R1 and R2. This R1 and R2 should be disjoint, that means there should be any common pixels any common points in these two partitions R1 and R2. Then, if I define a predicate say p over a region Ri that should be true where this p is a logical predicate defined over the points in set Ri in this partition Ri.

So, for a single partition Ri this logical predicate p should be true. And the last property that must be followed is predicate over Ri union Rj, 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 or 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 two different regions Ri and Rj then the predicate over this combined set Ri union Rj 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 two 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 into a larger region based on some pre-defined criteria, and in our case this pre-defined criteria 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 for which the predicate value is, or for all of which the predicate is true.

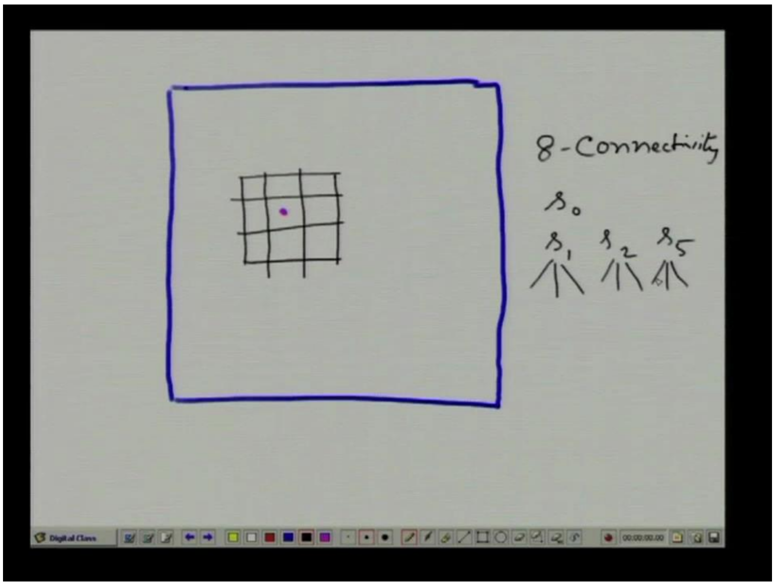
(Refer Slide Time: 41:37)



So, this region growing operation works like this, 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, this similarity measure can be different types, for example, we can say that two points are similar if their intensity values are very close and the point are dissimilar if their intensity values are, widely different and one of the condition that we have said that the points must be connected 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 same point.

(Refer Slide Time: 42:51)



So, for this purpose, what I will do is I can define I can have a 3 by 3 neighborhood around this seed point, and since, one of the property that these 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 these 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 star growing this region from the points which are connected to this seed point.

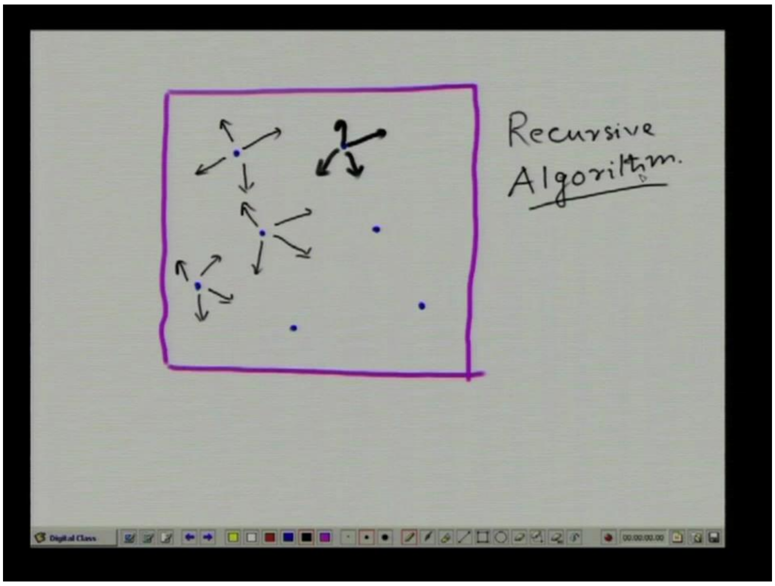
So, here if I use say, 8 connectivity, the concept of 8 connectivity. In that case, the point which are to be grouped or in the same group as the seed point they have to be one of, they must belong to this 3 by 3 neighborhood of this particular seed point. So, effectively what I am doing is, once I choose a seed point I check the points in its 3 by 3 neighborhood and all the points which are find are similar to this seed point, those points are 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 say seed point s zero. Now, from it is neighborhood I may find that the other points which can be put in the same group as point as this initial seed point s0 or say s1, s2 and say s5.

Next time I start growing the region from s1 it itself. I find within the neighborhood of s1, within this 3 by 3 neighborhood of s1 following the same 8 connectivity, what are the points which are similar to s1 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 s1, I start growing the region from s2, I start growing the region from s5 and so on. And this process will stop when no more new point can be included in the same group. Okay. So, effectively what we are doing is, we are selecting a number of seed points in the image following some criteria.

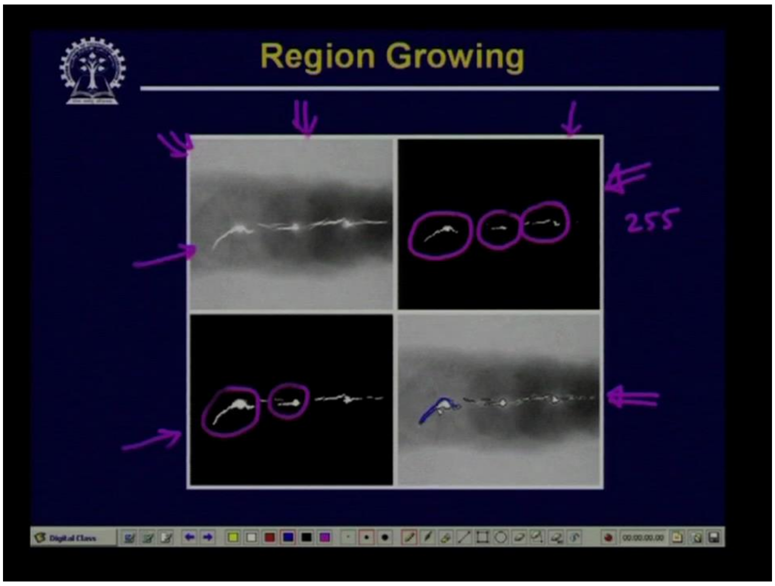
(Refer Slide Time: 45:58)



So, we have selected a number of seed points. So, the seed point selection is application dependent. And once we 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 region which are grown around this seed points. Okay.

So, this is what is the basic region growing operation and you 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: 47:17)



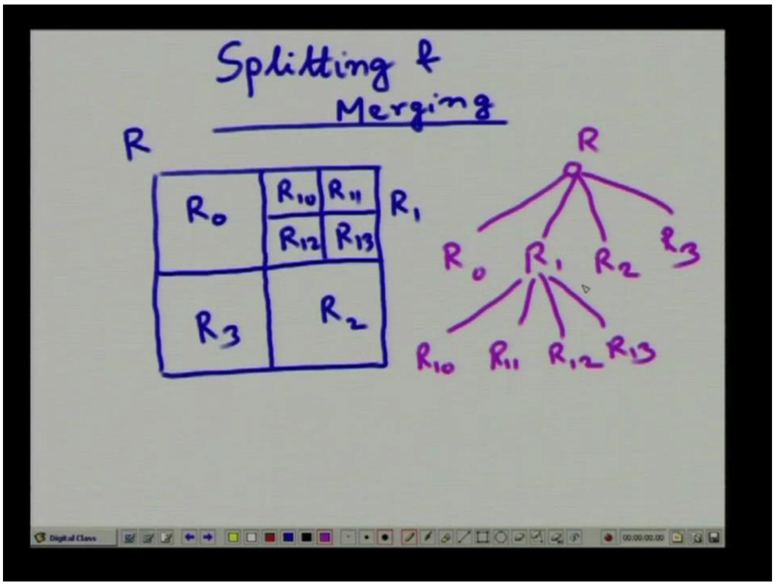
So, here is an example, this is the x-ray image taken from a weld. And you find that, in case of this x-ray image, there may be some cracks within the welded region or there may be some faults within the welded region, and these faults can be captured by using x-ray image.

So, this top left one, this is the x-ray image of an 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 based segmentation operation, where these are the pixels values, these are the regions having pixel values near and intensity value of 255 that is the maximum intensity level. 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. The thresholding operation is simply done to select the seed points. Once you get the seed point come to the corresponding seed point location in your original x-ray image and grow the regions starting from those seed locations within the original x-ray image. And this one shows the grown regions and now, we find that if I superimpose the boundary of these grown regions each of them are the grown region.

So, if I superimpose the boundary of this grown region on this original x-ray image, this superposition output is shown on the bottom right image. Here, you find that these are actually the boundary regions, boundaries which are superimposed on the original image. So, we 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. Okay.

(Refer Slide Time: 50:18)

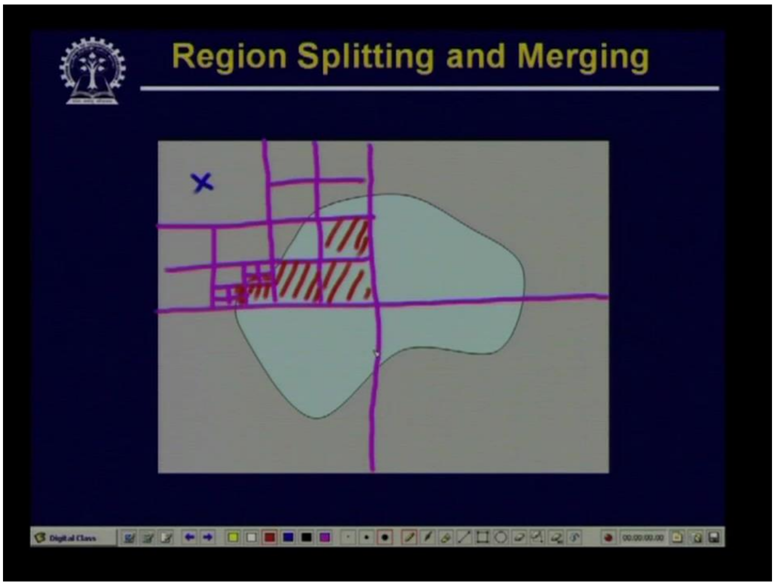


So, the next type of segmentation that we said that we will discuss about is, splitting and merging operation, Splitting and merging. Here again since, what we are trying to do is, we are making a trying to form a segment of all the pixels which are similar in intensity values or similar in some sense. Our approach in this particular case will be that if I have an image say R, first you find 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 four 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 is 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 R0, this was region say R1, this was region say R2, this was region say R3. Now, this R1 is non-uniform. So, I partition that again making it R10, R11, R12 and say R13 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 Quad tree representation of the image.

So, in case of Quad tree representation, if root node is R, my initial partition gives me four nodes R0, R1, R2 and R3. Then R1 I am partitioning again in R10, R11, R12 and R13. Once such partitioning is complete 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.

(Refer Slide Time: 53:14)



Now, let us see this with the help of an example, say I have an image of this form. When you come to this original image you find that here I have this background and on this background I have this object region. This is obviously non-uniform. So, what I do is I partition them into four 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 find that this particular partition is uniform, so I do not partition it anymore. The rest of the partitions I have to go for sub partition 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 anymore partitioning either I have this, a minimum partition size or every partition has become uniform, then I have to go 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 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, it can be combined with this partition and so on. So, finally what I get is, after the 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, a different object can be segmented out from the background. So, in brief we have discussed about the different segmentation operations. Initially we started with discontinuity based segmentation where you 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 we have discussed about various thresholding operations, the region growing operation and lastly the splitting and merging operations. Thank you.