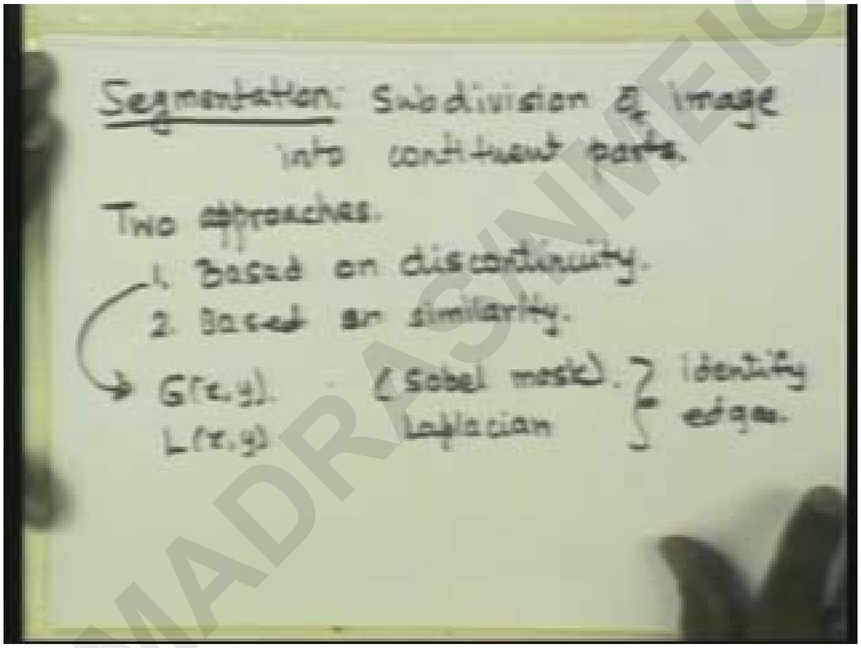
ROBOTICS

Prof. B.Seth
Department of Mechanical Engineering
IIT Bombay

Lecture No-30

Image Processing

Good morning from the last lecture we looked at how we could high light the edges okay we have seen different techniques for enhancing the image and one of them was to highlight the edges because many times we want to look at the edges because they are the boundaries of the object and background of times and we would like to be able to isolate the objects from the background so today we will concentrate on this process which is called segmentation (refer slide time04:28)

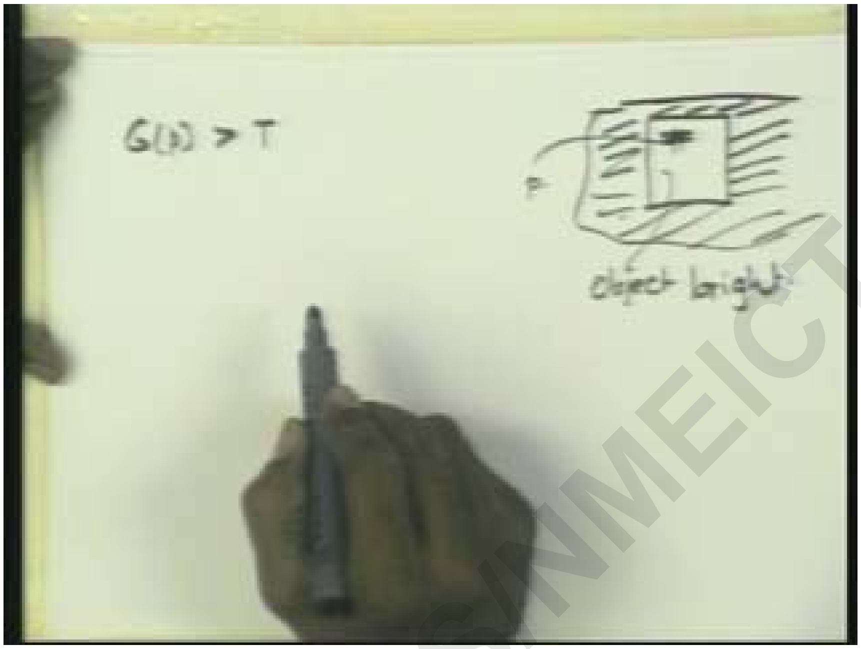


segmentation is basically sub division of image into its constituents or constituent sub images or parts which will be necessary for later on for recognizing the object which is the ultimate aim is to find you know what object is there and what other properties of this object how robot can interact with this particular object will be the ultimate aim so for that we need to first divide it into image into subdivisions which are hopefully corresponding to real differences between object and the background or object and different objects

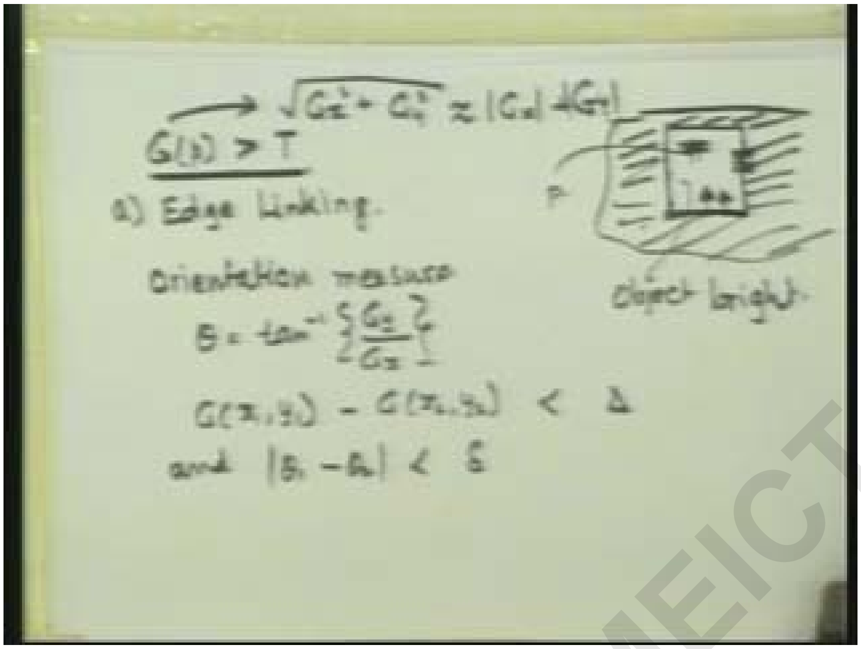
so there are two approaches basically which are useful segmentation okay the first approach is basically based on discontinuity and the second approach is based on similarity

okay so let us look at the first one based on discontinuity so we are looking at now edges will come out to be useful we have already seen how edges can be determined and so if you look at it we defined a gradient operator

okay which was basically also sobel mask does that job very well okay we have also looked at laplacian mask okay these two can be used to identify edges
okay let us now just look at that supposing I have the object which is bright and say the background which is which is dark okay object is bright (refer slide time05:40)



now one what one can do is first of all one can apply the sobel operator to this image okay so you will get the various points in the image say this is point p I can find out at point p what is the gradient of the image and if the gradient is large enough okay so I can compute g at point p and if it is greater than some threshold value
some threshold number that because what you remember is that gradient operator when the intensity changes suddenly gives you a large value if the intensity is uniform then it does not give you a very large value it gives you a close to zero
right ofcourse there will be variation from one point to another point some small variations will take place continuously because of diffused lighting
so but those will give out to be very small numbers but at the edge anywhere near the boundary if you have if you take that mask and run that mask
then you will find that the value comes out to be very large
okay so coming back here if we have the gradient determined at each of the points and if it is greater than some threshold then this is slightly to be a boundary pixel then okay now what we want to do is we want to join all the boundary pixels together
okay what will happen is(refer slide time09:47)



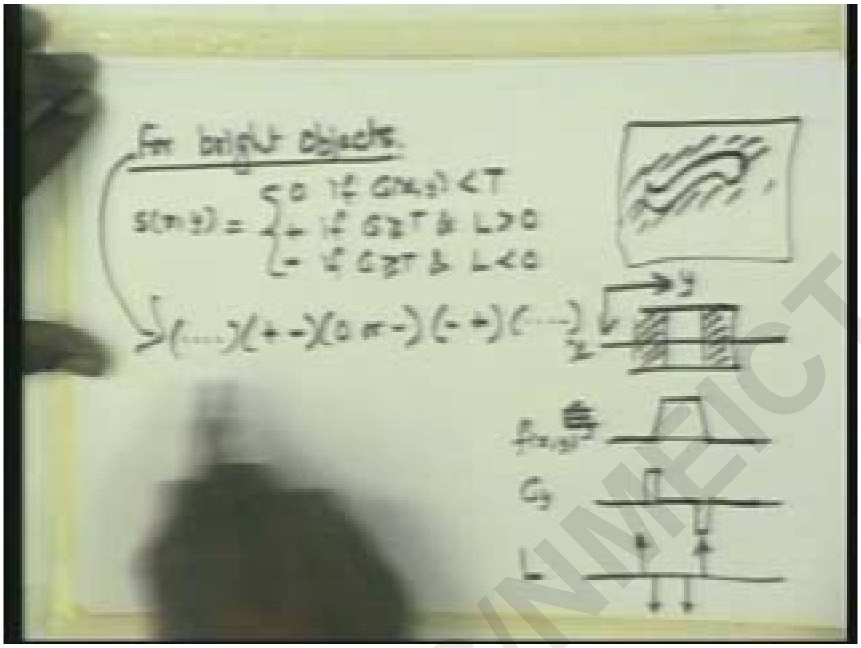
because of variation in intensities we will not always get all the boundary pixels so there will be some missing parts or we will get to many boundary pixels near the boundaries or there is a shadow so there can be number of effects like that so one technique is to link the edge so it is called edge linking okay obviously as the name suggest we are going to link up the edge pixels so what we do is we look at what are the similar you know which are the candidates for for forming the edge so what we do is we look at two things we look at the one is the intensity or sorry the value of the gradient and one is the orientation of the gradient so value of the gradient is what I have said and then orientation can also be found [noise] is some theta which is tangent inverse of y direction divided by the gradient in the x direction right this gradient [noise] about is say g x squared plus g y squared square root or it could be or you could approximate this as g x plus g y absolute values right number of ways you can define what the gradient value at a particular point is going to depend on the change of intensity in the x direction and the y direction because edges are not always going to be aligned to x or y directions so we will have to need to know the value of the magnitude of the gradient okay

so the what we will simply will can say is that if g of x one y one which is one of the pixels which is likely to be a boundary pixel this minus x two y two another neighbouring point okay if this is less than some number okay

so some delta lets say if this is less than some delta then what we are saying that is that the intensity of the edge is almost the same that means that it is likely belonging to the same edge so that is one condition

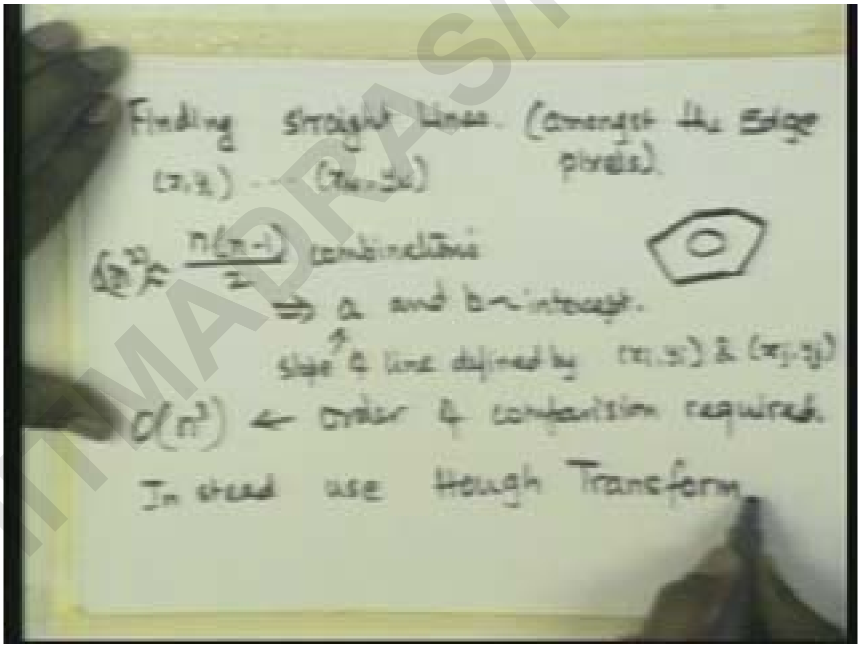
and that theta one lets say minus theta two absolute value of this is less than some delta value so if you look at which is point [noise] along the edges what this theta is of course [noise] various points and this point will also [noise] direction so that's why we don't care whether it is along the edge or not but it is connected to the edge and basically if the gradient direction and the magnitudes are similar then we will say they belong to the same edge so we link these together we will put it in the same set

So that’s one way of doing it now the another possible way of doing this is to also use laplacian operator okay so what we can do is
Some or let me(refer slide time13:35)



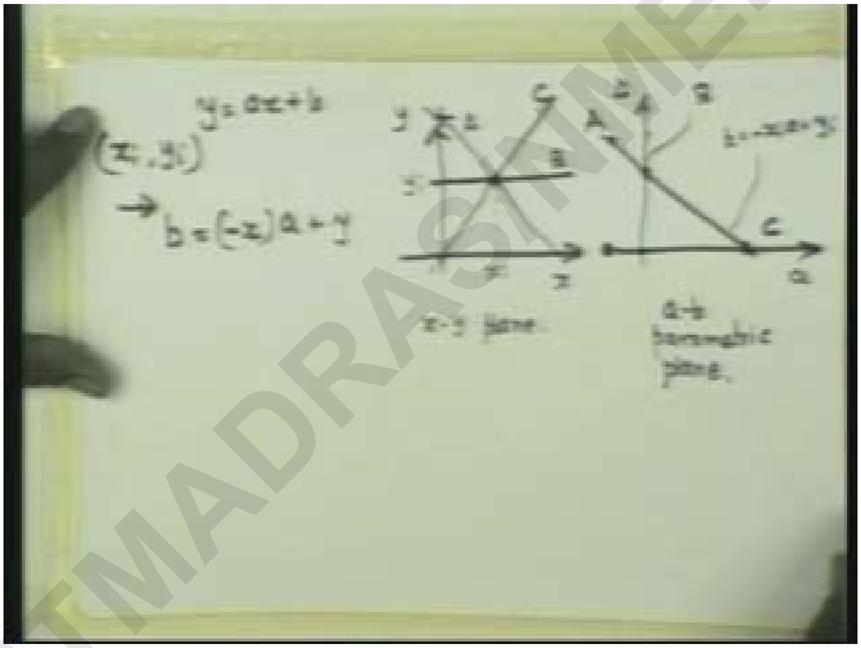
make the of a brighter object in darker background so if I look at simple situation like this where you have dark areas and a bright object then we know that the if I look at the gradient along the y direction if this is my y direction this is my x direction then g y is going to vary as okay first of all how the intensity are varying so lets look at f x y intensities as I go along the y direction say I take a line and I go along that and then I will see that the intensities are low at some point the intensity will raise and then have some high values so intensities are like that so therefore the gradient in the y direction is going to be looking like zero and there will be a peak here [noise] and negative kind of a peak existing here
okay if I look at the laplacian operator that is going to give me the second derivative and that will show me a high value here then high negative value here high negative value here and high positive value here okay so this will be the laplacian of the laplacian operator result now in c is that if for bright objects okay the sequence that I expect is that now first of all let me let me define three levels of intensities so first of all I will say that the transformed image s x y I am going to say that it is zero if my gradient is small enough if it is less than some threshold and it is positive for some grey level if g is greater than or greater than equal to threshold
and okay if the laplacian is greater than zero similarly I will call it a third level or minus level if gradient is greater than the threshold and laplacian is less than zero
right so I am now distinguishing different parts of the edge you know as I scan through along the x direction or y direction I will find a pattern like so for this bright object in dark background now what I expect is that there will be something
then there will be a plus followed by a minus okay then there will be zeroes or minus possibly and then I will have a minus followed by a plus and then I will have whatever

else right this is typically what I will expect if I have a bright object in dark background if I go along x direction or if I go along y direction
now this will help me to remove see what happens is that when we actually do this these processing in real images then we find that the effect of noise is that we will find spurious you know age pixels which are not actually there but they are because they are some kind of reflection has come and some problem in sensing is there there are noises present
now we want to remove these noises and say zero in on the edges which are belonging to the boundaries of the object to the background that is the purpose so
one way we saw was that we can link edges which are having similar gradients and which are having similar directions
right once then I know that pretty much it must be belonging to that same edge alternately since I know this is the kind of sequence that I expect if I divide the image into three grey level some you know some value some another value another value then I know that I expect the sequence
so I will be able to say that the edge belongs here and here this is the object and this is the background
okay if few noisy pixels here and there are not going to change my the result that I am getting here my conclusion here
so that is the basic idea okay now often times we are interested in [noise] in the image(refer slide time18:13)



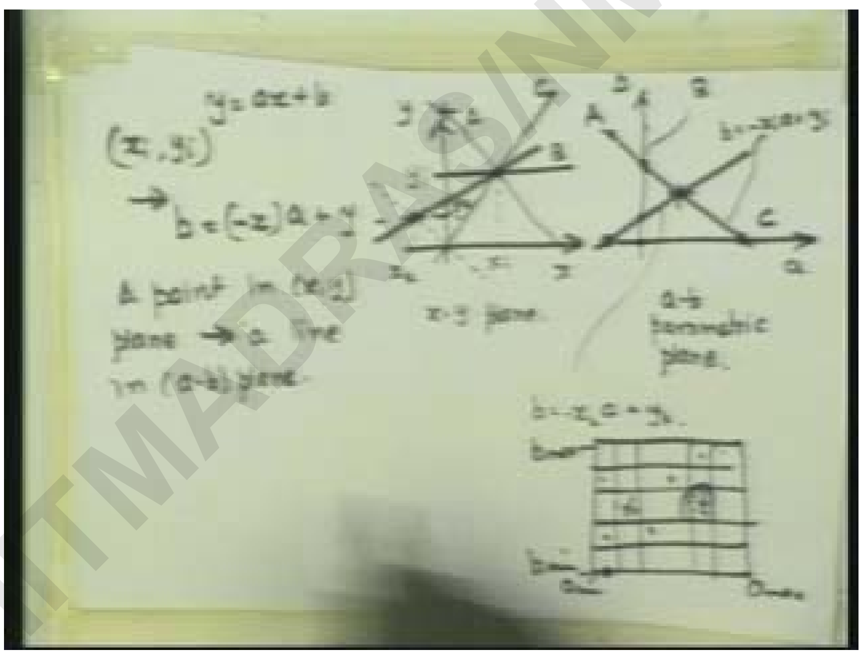
okay so straight lines amongst edge pixels okay why would we want to do that supposing there is a box shape you know object that is not lieing on the table
it has some straight edges okay may be there is some kind of a object like this so we would like to know what is the orientation of this so we would like to know where if there are straight edges present okay
if there are straight edges present how do we find one way would be to go and see okay um x one y one [noise] etcetera y n these are my probable edge pixels which are

determined because I use sobel operator and then I say okay I take pairs of these so from x one y one x two y two I can find out the slope and the intercept of a line which passes through them right I can find out from all these n times n minus one divided by two
okay combinations which will result in some value of the slope lets say call it a and intercept lets call that b so this is slope of line okay defined by x I y I and x j y j
similarly this is the intercept value
now what I need to do is given one line that I have defined using two points which are you know possible edge pixels then I would compare it with all other x and y values and see whether they fit also with the same line right so this way [noise] I will approximate
if I do the comparisons with all other points then it becomes n cubed order right there are comparisons or evaluations required
now this number tends to be pretty large so what one does is one can use instead a technique which has been hough okay and it is called hough transform
okay what is hough transform let us try to understand that
now let us look at [noise] and say I have a point here(refer slide time22:11)



with some coordinates x I and y I I can pass number of lines through this point right if I pass a line like that then I know that it has some slope and it has some [noise]
okay so this is lets say line a and now if I look at the plane which is defined as this was defined by x and y these are regular addition plane if I define a plane which is defined by a which is the slope and b which is the intercept then I will call this as a b parametric plane
okay this is my x y regular plane okay what I am saying is that given this particular line a passing through the point x I y I the slope that I have is a negative slope here
okay so I will have some value here lets say negative slope and some intercept which is positive so I have a point some where here corresponding to this line I have a point so I will call that a right

now I take another line let me take a line which is say horizontal passing through the same point x I y I
so I have slope is zero so I have slope zero intercept some positive number intercept some positive number okay so I get another point lets say this is b
let me take a line lets say passing through the origin so let me call this c now c is got intercept of zero because passing through the origin right so it is and it is got some slope which is positive so infact what I can show is this is point c is
right because my line okay if I am given a point here I am going to be able to write this parameteric equation of that particular point xi yi so xi yi point okay gives raise to or can be represented as a line in the parametric plane okay so this is going to be given by in fact okay will be given by because the equations of the line of a line is a x plus b now what I am all I am doing is simply rewriting in a different way so I will call this y not I write it as b is equal to right minus x times a plus y it is the same equation I have not transformed it and what I am saying now is now I substitute for x I y I so I this equation that I have got here okay is nothing but b is equal to minus x i times a plus y I
okay interesting thing is that each point in the x y plane corresponds to a line in the parametric plane
okay if I take some other point so let me take one more point here (refer slide time26:57)



so this is my x k [noise] once again I can find infinity of lines passing through this point and they will all fall along a straight line here
okay as it turns out I will get say lets my zero intercept here will be a line which will be let me show it as dotted line [noise] slope and zero intercept right I have a horizontal line which is got zero slope see this is got zero intercept this is zero slope and [noise] here ok and likewise I can this was that and this line here is b is equal to x k minus x k times a plus y k so every point a point in x y plane transforms into okay a line in a b plane or infinity of points in a b plane

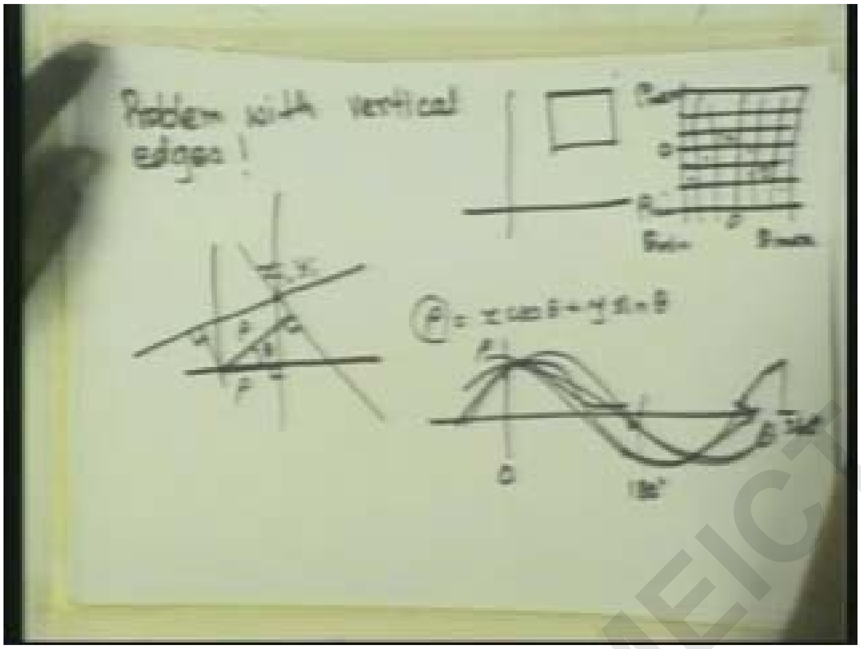
now look at this we have two points and these correspond to two lines and they intersect at this point right this is nothing but the line which is common to both x I y I and x k y k so this infact represent the line joining passing through both these points

now we will use this idea to say that if I have a image and I have already identified some boundary pixels so from those boundary pixels if I take one by one each of these boundary pixels and apply this transformation to get the corresponding line in the a b plane I am going to take each of the boundary pixel x y location and find corresponding to that the a and b values then what I will get is I will have a image now I will transform this into some graph or infact what I am going to do is to discretely divide my a and b value so I have some a minimum and a maximum okay b minimum and b maximum and I have compartmentalize this div divided this values meaning I am discretizing slope I am saying okay any slope which is between zero and say ten decreases going to be one category

ten degrees to twenty degrees will be another category similarly I will say that every intercept which is zero to one will be one category one to two will be or whatever the interval I choose depending on what is the kind of accuracy I want in detecting or finding straight lines in the image then for given point which is a boundary pixel okay I am going to take some value of a mean find out the corresponding value of b mean whatever that corresponds to I will put a dot in this particular box then I will go with the next value of this thing so I will get various dots in these boxes

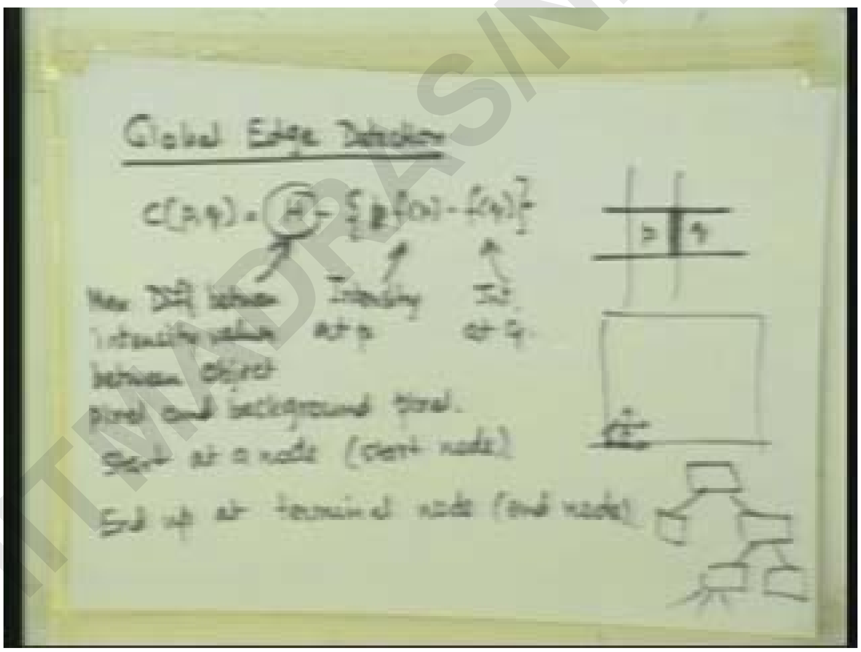
okay I will take then one more point and find out the whatever dots come into these boxes so now I am I am doing for fewer calculations because depending on how finely I have resolved this a and b parametric plane I will find that there are some boxes which are heavily dotted and some are sparcely dotted okay so what the meaning of this heavily dotted box here will be that this value of a and b is actually common to a whole lot of points which means a point in the parametric plane actually corresponds to a line in the x y plane so this means there are whole lot of points which belong to the which lie on the same straight line and this is basically the approach of the hough transform that you discretize a and b parameter then you find out where the maximum number of points are those are corresponding to strong or large straight segments one problem that one encounters in this approach is that if you have a vertical line if you have a vertical edge in a picture then we cannot easily find the a b parameters because both a and b are infinity for the vertical edge the intercept is going to be infinity and slope is also infinity

so we run into a problem with okay problem with vertical edges (refer slide time31:56)



one can get around this problem by simply saying that okay let me define a different type of transformation its also hough transformation so I take a point now instead of finding a straight line parametrization a b parametrization I will [noise] parametrization now supposing I pass a line through this point this is what I have been doing then I will say okay let me now characterize this particular line by a perpendicular drop from the origin on to the line okay this is now going to be characterized by some theta and some row value I can pass another line through the same point and
I will find that I have parametrization [noise] okay this will get around the problem that if there are vertical edges there is no problem now because if I take a line vertical line passing through this okay
the perpendicular drop will be this and this will be the row and theta will be zero similarly horizontal line is no problem so the relationship between row and theta and x y okay simply can be written as you can show geometrically x times cosine of theta plus y times sine of theta okay so what this means is that given a particular point row theta parametric plane now I am going to have some kind of a sinusoidal curve
okay depending on the value x and y if I thet change theta from the range of zero to two hundred three hundred sixty degrees then I will find that this is going to be a repeating function infact I I don't need to go to upto three hundred and sixty degrees I can chop off I can take the area from zero to hundred eighty degrees because after that curve is going to be predictable
i know this much information I can I can do the rest of it also it is no problem so what is now going to happen is my transform is now going to be such that corresponding to each of these values of x and y which are belonging to the boundary pixel once again I am going to have my parametric plane in which I am going to have theta mean theta max
slightly to be from zero to hundred eighty degrees and row max okay or I can take from minus ninty to plus ninty so[noise]

now I sub divide that region parametric plane into boxes depending on what accuracies once again I am looking for and I carry out the same procedure that given a boundary pixel x I y I
i take various values of theta depending on how finely I have resolved it and for each value of theta I can correspondingly find out the value of row and I point for each value of theta I then plot this and i I am expecting to get sinusoidal type of variation here approx
and once again for different points that I take I am going to get different sorry different sinusoidals and whatever number of sinusoidals I will find that there are some points which are intersecting these curves different curves correspondingly I will get once again different number of points in the same box
Right so this way once again I can find out which are the points which lie on the same straight line the value of the slope etcetera that can be known from the value of theta and row so this way I can I have to if I don't have very fine resolution here then it gives me a advantage in terms of number of computations or comparisons which are required
Okay now all these procedures that I am describing right now I mean they are quite prone to noise that are present in the picture
So let us [noise] scheme which is immune from noise so this I will call(refer slide time36:45)



Okay Idea here is that we define a cost function so if I have say two pixels say p and q if I go from p to q then what is the cost associated this is something that I want to give and my cost p comma q equals to some value minus the difference between f p [noise]
Okay then this is intensity at p this is obviously intensity at q and q is a neighborhood pixel p could consider four neighbors okay so this is the four neighbor here and what I will choose this h to be typically the difference between intensity values or maximum difference between intensity values between object pixel and background pixel

Okay so supposing the edge was passing through here this was this belong to the object and this belong to the background or viceversa whatever then what will happen is this f p minus f q would be a large number as large as h right or at the same ball part figure order of magnitude and when you subtract out this difference from this constant value h that I have taken I will get a very low cost going from p to q I have low cost [noise] here because [noise] between them if the [noise] boundary exist It was the same background lets say then f p and f q are going to be similar values

So this difference is almost zero therefore I will get a large cost so I am iam just saying simply defining the cost function between two pixels and what I need to do now is to say I have a start node okay so now the procedure is start at a node called a start node or a beginning node and then I have a terminal node

We want to end up at terminal or end node and I want to find out the cost of different parts that I take from beginning to end so if I have now some image I say start from this pixel then I can go this happens to be corner pixel so I can go this way or that way

I have only two possibilities then I can go from here in two directions once again this direction I need not go because I have already covered that but once I come here I have three directions to go to right so I am going to get a tree structure depending on how I start I start here and then I get two possible places and then I get two other possible places similarly and then after that I may have three possible places so I am going to get a tree structure and this tree structure will be fairly dense depending on how many pixels that I have to cover if the image is you know five hundred by five hundred then the problem with this technique is that it is very very intensive in terms of computation the combinations that you have to consider of different paths basically we are looking at all possible combinations of paths going from one point to another point

so there is going to be a very large number of such paths so whether I go this way this way this way this way or I go this way this way this way this way you know you know any number of such paths are available from going from one point to another point

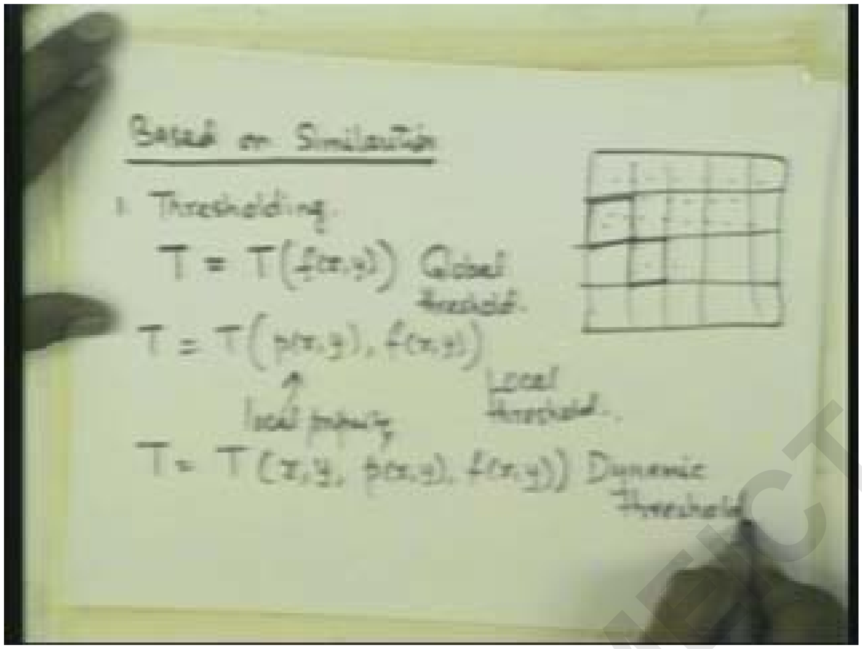
and I will compute the cost for each of these paths and I will say which is whichever is the minimum path is the best boundary okay so

advantage here is that I am not very much dependent on one or two pixels being wrong because overall the cost is going to be turn out will turn out to be the lowest if I am generally going along the boundary

okay but as I said the disadvantage is that there is not too many parts of it so one will have one actually does instead of getting the optimal path one sacrifices that and tries to have some kind of a trade off between how much time it takes to compute it and whether we are going to be very very optimal or not so one one can have teams in which one looks suboptimal sub regions where you have found out some optimal path then you assume that whatever is optimal is not going to change because of rest of the image

which is not strictly true okay but the combination of local and global can reduce the overall computation and give you fairly good results

okay then let us look at um see we have been looking at boundary based segmentation we can look at now the second type of segmentation which is based on similarities [noise] (refer slide time42:07)



okay in similarities first thing let me consider is thresholding okay thresholding is you will know is you will select a critical value of intensity and then anything above the critical value will be taken as one region

everything below would be taken as another region so we are generally talking about images in which there are objects and we are not looking at surface feature so much as what is the boundary of the object versus the background

so if I define a threshold t and so one possibility is that t will depend on the intensity value or distribution in the whole of the image

so if I have some image I am not going to try to draw image actually but I have various intensities at various pixels

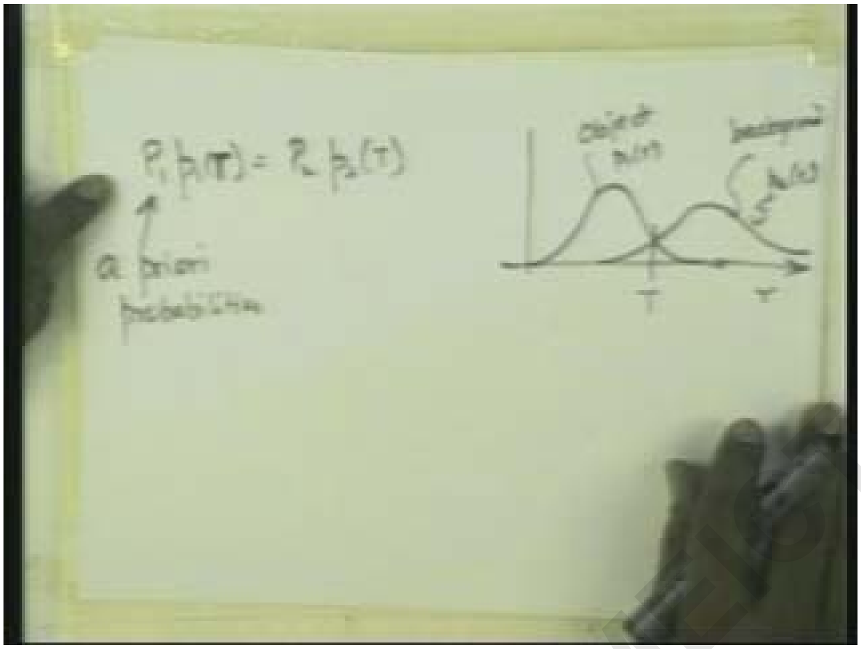
okay now depending on this intensity values I select some value

and say okay anything about that is going to be one everything below that is going to be zero so that I can threshold and this is called global threshold on the otherhand this technique is not going to work very well if you have different objects which are lying in different areas with some change in the background intensities there are shadows so what will happen is supposing I now look at sub images of the image now this is not actually a pixel there are many pixels inside each of these regions so I can do some local analysis here

say okay in this local region what is the best threshold in this local region what is the best threshold and these thresholds will vary depending on how the the theme is right so if I do that then I might the threshold becomes not just a function of the intensities but some properties that I am looking for the neighborhood of x y which I am calling as some local property and it will also ofcourse depend on the individual pixel intensity level to decide what is the threshold then this is called the local threshold

finally I can have a dynamic threshold which not only depends on these two but also depends on the individual pixel locations then this is going to be called a dynamic threshold

okay now for thresholding basically(refer slide time44:24)



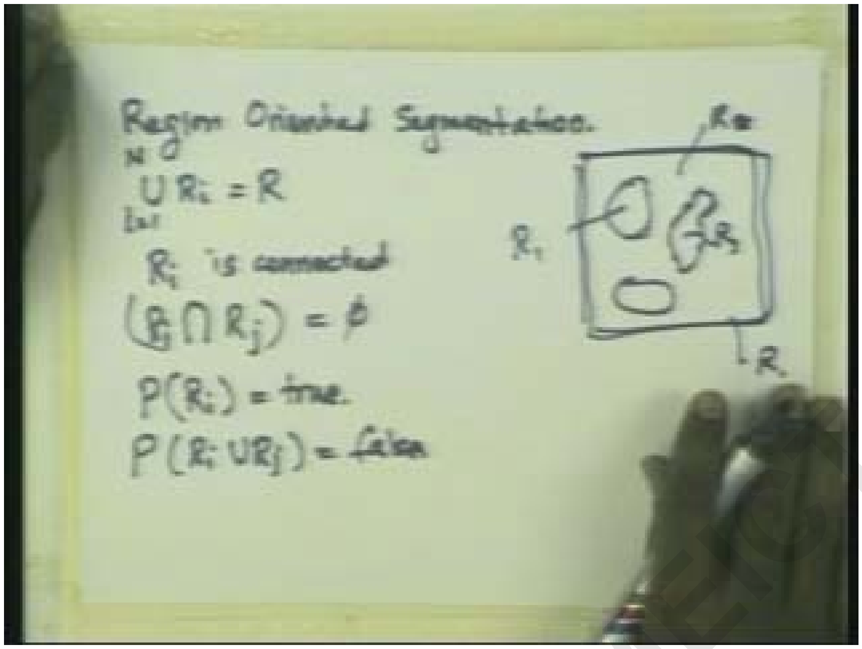
what we do is if I take a simpler situation if I look at the distribution of pixels in different intensity level so I look at the probability distribution function so then I will have say objects look like this and say the distribution function for the background say something else I will have some kind of a probability distribution functions corresponding to say object say this is the background right then what we do is we can select a threshold here which will be the best way to distinguish the object from the background

mathematically I can say that if p one times this is lets say p one of r r is the intensity value and this is the number of pixels in that intensity value times p one at threshold should be equal to p two capital p two into small p two at threshold this is small p now capital ps are what are called a priori probabilities okay this is known from the general situation supposing I have a conveyor belt in which you have some parts coming then I know that one part on the average comes every five seconds so given any scene I will have on the conveyor belt I will have three parts lets say area of the conveyor belt is so much area of the parts is so much at any given time I expect the parts which means you know corresponding number of pixels at any time are expected to be object pixels and the remaining are expected to be background pixels

if that is the case then I can multiply these with the probability distribution function of each of the objects and then I can find pick my threshold in this

one can go more and more in details of these but I think we will not have time to do all that so let us get some more flavour of some other techniques okay

so let us look at one more technique based on similarity [noise] region oriented segmentation (refer slide time46:51)

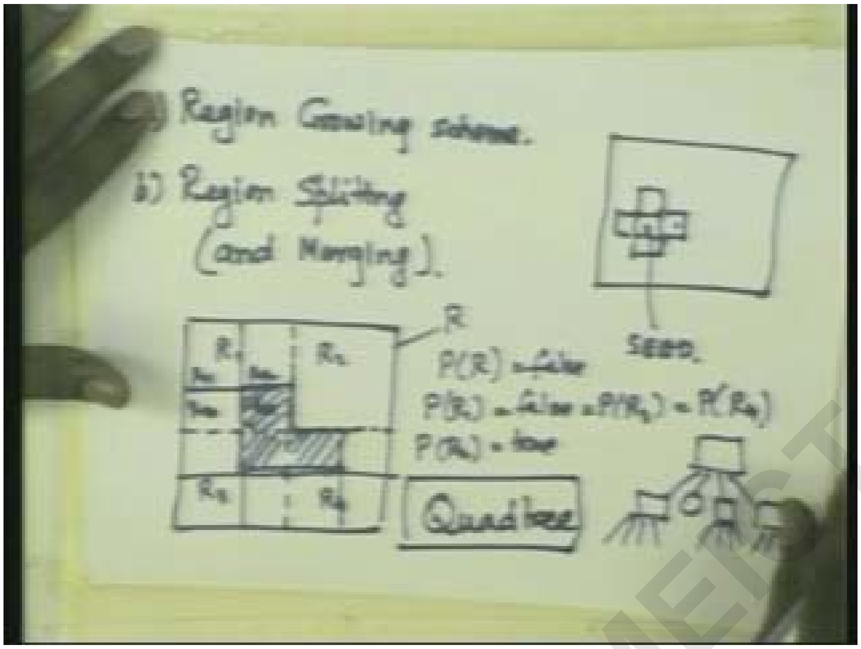


okay what we want is we have a image we have various objects in this we would like to divide we will have to define these as some sub regions okay so lets say this is r one this is r two this is r three and so on so so what we want is basically that union of all the regions I going from one to whatever number of regions are there should be the total region so let me say this is the total region

so let me say this is the total total is r okay so and I am going to define the region so that each of these r I is connected now definition of connected we have already seen in the introductory lectures where some kind of a property may be intensity is belonging in some range right so each of the region should be connected but if you take two regions [noise] p of I okay and all these things if I take [noise] p I then take intersection with r I and r j this should be zero there should be no pixels which are common to do two regions okay and then we should have some property so this is some property of region r I it should it should be true otherwise we will not be able to call it a region and similarly property of If I take two regions together then it should be false right so this this is what I am calling um u know sub image or region within a image which are defined or which are characterized by this type of a set of properties

okay now in one simple minded scheme what I can do is [noise] I can [noise]

(refer slide time51:55)



say if I am looking at four neighbors then I [noise] at this point okay if it is belonging it is having the property [noise] whatever the property I am going to define a region then I will say okay this is belonging to some r I this is called region scheme whereby I look at the neighboring pixels if the properties are similar then they belong to the same region and I keep on going until I encounter a boundary where the property is changed and therefore I will not be able to go any further I will not be able to grow the region any further then I will have to start with another seed point at a different part of the image and I start growing it all around the game and then so this way I will put the seeds and I will find out all the different regions in the image until I cannot find any new in there are no more pixels remaining of

another alternate scheme is what is called region splitting and let me say and merging

okay the basic idea here is let me illustrate that by taking an example say I have an object here lying in the image and say this is some intensities which are very different than the background I am not coloring it then it becomes very difficult to see

so what I do is I have this region as the whole image first I check for the property of this region r and I find that this is false what I mean by that is that if I look at various parts of this particular region I find that there are different properties there is no single property which is or whatever property I am considering is not uniform right so then I take this region and I subdivide it into sub regions say I divide into four equal sub regions and then I look at this is my r one this is my r two this is my r three this is my r four and I find out the properties at r one and I find this is false but property at r two is true because it is a uniform region okay all pixels have similar intensity values or properties and this is and similarly I will find that this is property at r three is also false property at r four is also false so what I am doing is I am starting out with the whole image then I have sub divided into four nodes r one is false so I need to subdivide this into four but r two is true so I don't have to subdivide it anymore where as the other r three and r four I will have to sub divide into smaller and smaller square so I have chosen so that these boundaries are matching okay they will not match which means will have to do further sub divisions right

so once I have done this now I have this as say r one one r one two r one three and r one four and each of these now have uniform properties okay because this has dark this has dark this is dark but all others are bright so we have

now I don't have to split anymore now my time for merging starts here so I will now consider r one one r one two which are neighboring regions and I find that they have the same properties so I remove this boundary and I define this as a region

similarly I will start looking at all these sub regions and I will find that the background is now one region and the object is another region

okay this is called quad tree

so I think we will have to stop more or less here what we have seen is in this part of the course we have seen that um is we can look at scenes we can acquire images then we can relate the pixels which are there in the camera frame okay camera frame of reference and relate them to the external world frame of reference

and we saw perspective transformation in that we have also seen stereo images how we can get information about the depth right the problem is that imaging transformation is basically a three dimensional transformation to two dimensional imaging transformation two dimensional image and therefore some information gets lost and the same pixel in the image corresponds to infinity of points in the real world and therefore we need some additional information when we want to go and reconstruct the world from the image

so either you have to have two images and need to know corresponding points okay we also look at the camera calibration problem

then we looked at number of image smoothing operations okay which is basically all part of pre processing we looked at some enhancement techniques which is basically highlighting the um we looked at histogram based approaches and we looked at edge enhancement approaches um then what we have looked at is various segmentation problem how to divide the image into various segments or regions which are corresponding to objects and backgrounds

now from here there is a whole lot of other things remaining which is that once you given a particular object how would you now recognize the object as a region as a object so one will have to now start defining properties for each of these objects each of these regions I should say um you can have area as a property you can have orientation you can define boxes which will enclose it so or you can define aspect ratio or you can define centroids you can define the shape in some way of how you have to traverse the boundary to enclose the object

so there are there are host of different properties that you can find out and then based on the properties one can determine what the object is so it leads to then the recognition of the object so If I am trying to compare say washers and bolts which are coming on the conveyor belt I have images of those then there are different properties which will distinguish this these so I need to define those properties depending on the object and then I can process the image find the region see look for those properties if the properties match then I have a good confidence in saying that this particular region belongs to a washer so this is a washer now I have recognized a washer I can find out other properties of that and then I can have tasks which are connected with robot movements which will then use this particular information that the object or a washer is present in such and such

location in the image which I know where it corresponds to in the real world because often times is that distance will be known if it is a conveyor belt and a camera hanging above then we know the distance between the camera and the plane in which the objects are there and therefore we know the z absolute global z location and therefore we can now connect the x and y location of the outside world from the image points so this is in a nut shell I have given you how image or machine vision can be useful for robotics
It is by no means exhaustive and there are so many things that one can talk about in this um but simply we don’t have you know time to cover all of those thank you