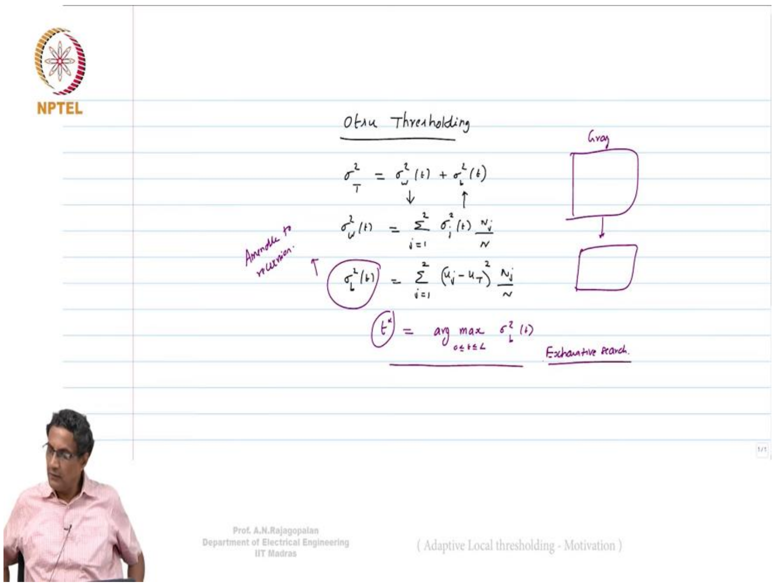
Image Signal Processing
Professor A.N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture 59
Adaptive Local thresholding - Motivation

(Refer Slide Time: 00:16)



So we saw that with respect to Otsu method... so Otsu which is a thresholding method - so we saw that for a threshold t, small t that we chose, we showed that sigma square t that is the total variance, which is of course independent of t, small t is equal to sigma square w of t plus sigma square b of t where we of course understand w to be the within class variance and b to be a between class variance.

Now, we want a t that should be optimal and therefore the optimal t, the optimal t, that you would like is a one that will either maximize between class or in effect, or in effect it will be, it will also minimize sigma square within class variance. Now if you look at the equations for sigma square w of t, what we had was summation j equal to 1 to 2. Sigma square j of t into nj by n. And for sigma square b of t, what we had was summation j equal to 1 to 2.

Then we had mu j minus mu t that is mu, that is total mean whole square into n j by n. And between these two like I said you know, one could, one could actually minimize this or maximize this; but between the two we typically end up choosing to maximize sigma square b of

t. So we actually maximize the between class variance and therefore, and why, why do we normally do that? It is because this involves only the means whereas sigma square w involves second order statistics which is variances.

And secondly, this sigma square b which is the between class variance is actually amenable to, it is amenable to a recursion which in turn reduces the, reduces the computational complexity, amenable to recursion. So what you are really looking at is to, is actually to solve for sum like t star which is the optimal value for t, small t, such that this is equal to r max, sigma sqaure b of t where this t runs from 0 less than or equal to t, less than or equal to l, right?

So if you see, this is an, this is an exhaustive search So what does this entails is an exhaustive search? Which is why time complexity, ti can be high because you need to search through all values of t and then right through this exhaustive search, you find out that value of t for which sigma square b is a maximum. And that would be your say t star.

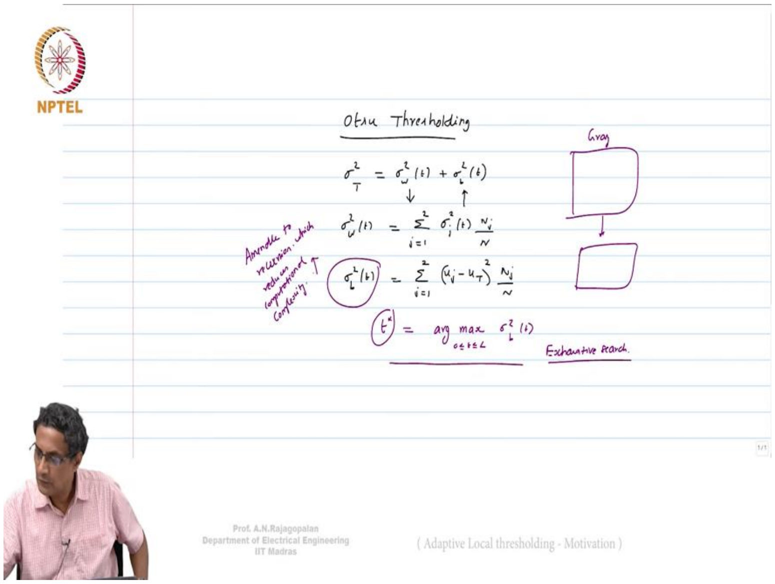
Okay, that is an Otsu thresholding, so if somebody gives you an image and, a gray scale image and if you want it to turn it into a binary and you want to look at what particular t star should you choose such that all intensity is less than or equal to t star belong to right and it can be reduced to gray level 0 and all intensities that are greater than t star are assigned the value 1, that you would actually find out. And this is one more thing, right?

(Refer Slide Time: 03:53)

Let me just show you an example. So here is a thresholding implementation. I will also explain what we mean by a global threshold. Okay, I have not actually told, I have not explained what that is yet. But you can see that here is a human face and this is a gray scale image and you want to convert it into a binary image.

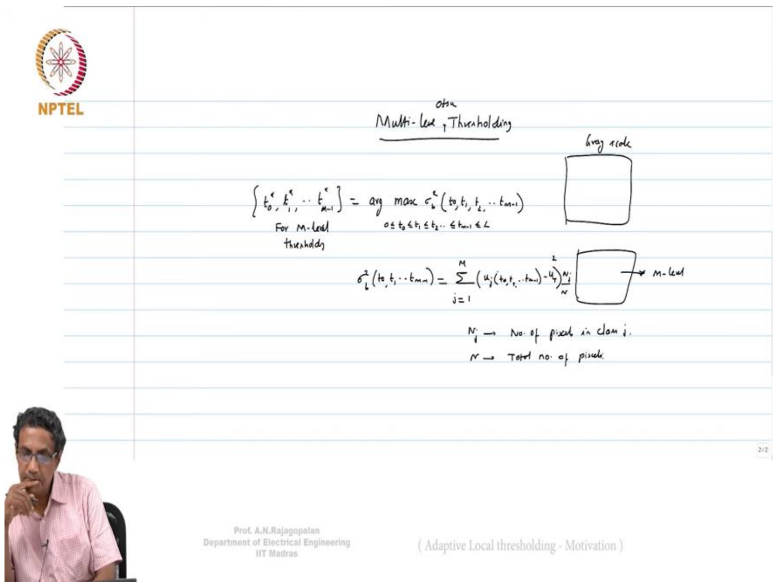
So you can see that it makes sense in the sense that, you know, pixels that are little dark; for example the hair and then the background, those have all been assigned the value 0. So they have all gotten to the black region and even the, even the beard and moustahce and all that - that has all come out as black or a 0. And then the face and the ear - those have come out to be 1. So that is how it works.

(Refer Slide Time: 04:44)



Now moving on, you might actually wonder now, so let me just add this point that it is amenable to recursion which reduces computational complexity, right - which reduces computational complexity. We will not really, we will not enter into what this is and so on. This helps to know that something like that is possible. Now at this point of time, you might ask how does this scheme extend to let us say mode levels - multiple levels - what is called multi-level Otsu Thresholding. So if you were interested in doing a multi-level Otsu, then that would be even more exhaustive as you can see.

(Refer Slide Time: 05:33)



And the way that works is - multilevel thresholding, so what that actually means is that you have again an image but then instead of just classifying - okay this could be again a gray scale image and you want to convert it into another wherein let us say you want to give, you want to assign maybe n number of levels, let us say four levels, five levels - whatever. If you want to go into some n level assignment. And if you wanted to see how you could extend Otsu to this, it is very straightforward.

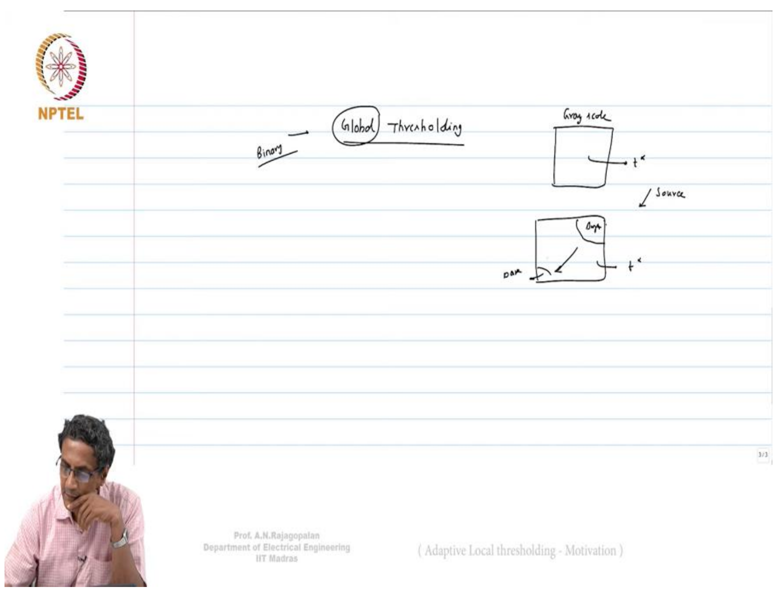
So you can again go ahead and solve this optimization problem which incidentally happens to be a global optima right? So you have to search for it but it is an exhaustive search. And what this means is that let us say that you are looking for let us say t knot star, t 1 star, all the way up to t m-1 star; because if you have m levels, this is for m level kind of a thresholding; m level thresholding. So you will need you know, t knot star, t 1 star right up to t star m minus 1. This you would find as arg max right?

Argument of course, you know? Argument that leads to maximum of this function, right? This is what you want. You want the argument, you do not want the value of the function, you want the argument that actually maximizes the function. And this is like sigma square b but now this is going to be a function of t knot, t 1, t2, all the way to t m minus 1. And this you will maximize over 0 less than or equal to t knot, less than or equal to t 1, less than or equal to t2, all the way up to less than or equal to t m minus 1, less than or equal to l.

So this again an exhaustive search and therefore this will of course become even more computationally complex. We will not actually delve into details of this. This helps to know that Otsu naturally extends and where this sigma square b. Of whatever t knot, t 1 and all that we have here, right up to t m minus 1, will now be given by, given as summation j is equal to 1 to m number of classes.

And you have got like mu j of now, when I say, I really mean t knot, t 1 all of that, m minus 1 minus mu t, the whole square into n j by n - where n j all means the same things, right? Whatever we had earlier, so those things remain the same. So n j, for example, would mean number of pixels in class j, pixels, number of pixles in class j and so on. And n will be total number of pixels, total number of the pixels in the image. And mu j right, again will be a function of right all these values, threshold values. And okay.

(Refer Slide Time: 09:25)



Now after this, this is one more thing that we would like to, I would like to talk about. So earlier, you saw in that example that there is something called a global thresholding - is what was written in that image, right? When I flashed that image to you. Okay, what was written was a gobal thresholding? Now what do we actually mean by that?

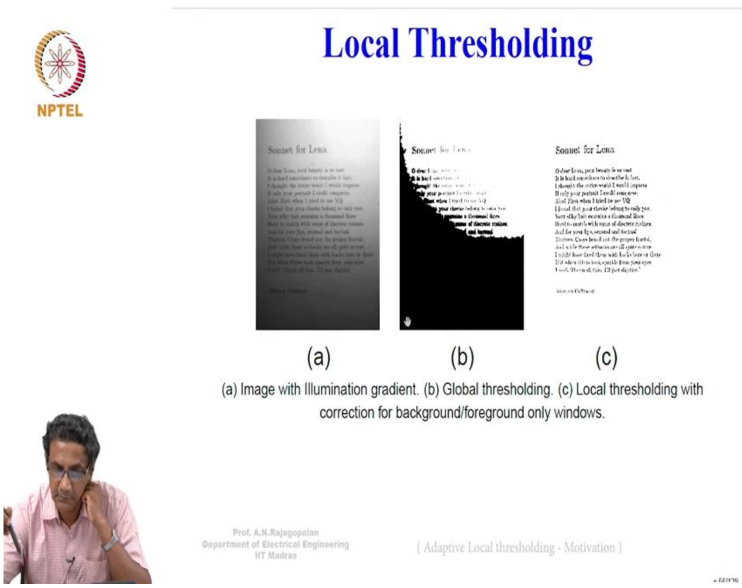
See, what we have implemented right now, what we showed is the example and what we actually explained is an algorithm, was actually for something called global thresholding which means that given this gray scale image, given this gray scale image - for the entire - so suppose we again, again go back to the binary case, two level thresholding, let us just keep this simple.

So for the two level case, what it meant was - you picked up a t star, which was the most optimal. But this t star is supposed to work for the entire image? That is why we call it a global threshold; because it is supposed to hold good for the entire image. But then is this always true, is it always true that, you know, you can actually pick up one t star and say that it works for the entire image? The answer is no. For example, right, you might have an image that has some a gradient in terms of the illumination.

Typically, when you see an image, you seem to sort of assume that it will have an illumination, you know, that is uniform. But it need not be the case, right? Sometimes image that you have a light source, that it is pointing here and therefore, right, it will tend to illuminate this part of the image more and then as you come towards the other end of the image, you might see the affected of the illumination go down. So if this gradient was not there in the sense that this illumination was actually a constant, then maybe it would have made sense to pick up a global t star.

But now given that illumination has a role to play and therefore, right, there is a gradient in there, what it could it mean is - if you, if you now computed a threshold based upon the entire image and then you came up with some t star, and suppose this side is bright, and this this side is relatively dark, right because the illumination, because the source, the lighting source is on this side, then what it would actually mean is that this t star, right, so you might have let us say, let me just show this image again, because then you would understand better.

(Refer Slide Time: 11:57)



Now this was local case, now this is, that was a global case; this is a local case. Now imagine, this is actually a Sonnet for Lena? Lena who you know quite well by now. So this is Sonnet for Lena and you can see that this is actually a binary text, right? There is a white background and then on which there are these letters in black. Now you would think that if I do a binary ration, I should get all these letter out ideally, and then the background should just become either white or black.

So the background should probably be white and then whatever is my text that I have actually written or the Sonnet should come out as black. Now if you just now look at this kind of a gradient in the illumination, right, you find that you know this portion is more illuminated as compared to this. And this is not because we underlined information is like that, it is being influenced by the, you know, by the illumination.

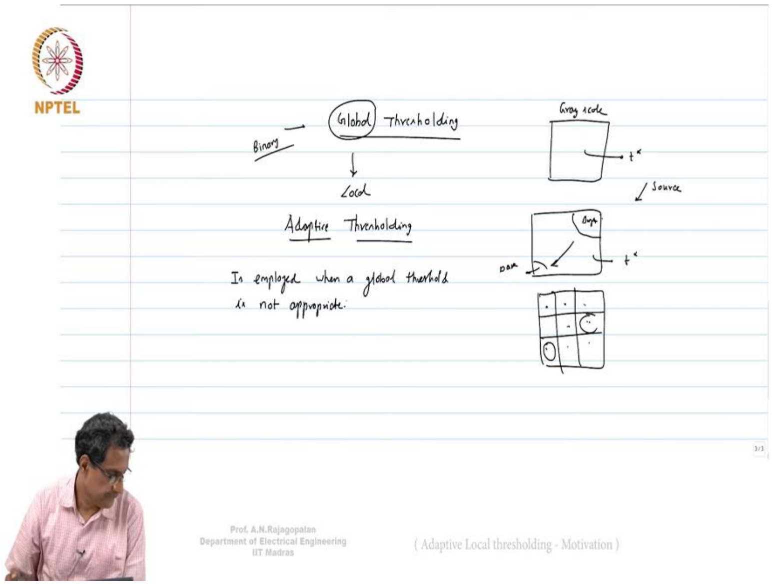
And therefore what this means is that, if you try to compute a global threshold, t star, use the whole image and compute it, then that t star may turn out to be, may turn out to be less for this portion which means that you know, here where there is a black text, but then because illumination is higher there, this may all be assigned a value 1 - in which case you might lose all this information - all the text here, which is what you see here. So the b part is in fact a global threshold, right? And there it shows the effect of a global threshold as you can clearly see that they have lost out on all these written words.

On the contrary, when you come here, right, in contrast, when you come down here - because this is already dark, right, and you know this is more dark, therefore what will happen is the t star that you have picked up, right, is probably you know, is probably too high for this region and therefore, even things that you would like to be assigned as white, for example the background will all become 0 and therefore you end up, you know, losing all of this information, information. Right, even these words are gone because these letters are also gone.

And what has happened is, the entire thing got kind of merged, right? The background which should have been ideally white got merged with what is black and all the red in between values that supposedly should have been white also became black because of the illumination gradient and therefore, we lost everything here. Now, if you try something like a local thresholding which I have not explained till now, I have not even talked about how this local thresholding works. I am going to talk about that scheme.

But if you had a local thresholding scheme then of course, right, you could get something out like that which makes a lot more sense because you have got all this text out against a white background. This is what, well you could of course improve up in this - like I said you know, this is with respect to some kind of an optimality criterion. But whatever it is, this is definitely better than getting something like this?

(Refer Slide Time: 15:09)



This is what, this is the, this is sort of the motivation for doing a thresholding that we say is local. That means, instead of finding a, finding a global threshold, right, we might want to find out, you know, maybe, in a sense what you are looking for is in the image, I should probably have a threshold, you know, for this region, for this region, a separate threshold for this region, a separate threshold. So it will like maybe, you know, for this I need a different threshold, for this I need a different threshold and for this I need a different threshold and so on.

And then again, right, at a finer scale you might actually want to know what should be, in fact, you might even go to the extent of saying that I might even need, you know, a threshold, I mean so what, now again you might ask - is it alright to use this threshold for the entire region, or should I even ask for, you know, for a threshold for each of these (())(15:59) cells. But you can go down to that level, right, then you can, at an extreme you can ask for a threshold for each pixel. And you might want to say for this pixel, if I have a t star, right, if my intensity is less than t star then it becomes 0, if it is greater than t star, right, then you know, then it would become 1.

So you could even stretch it to that extreme. On one extreme, you have a global threshold which is like one threshold for the entire image. On the other hand, you can have a region where you sort out the threshold and at the other extreme, you can have a threshold for each pixel. Now how do we do this? Now, there right, I am going to talk about something called adaptive thresholding.

And I am going to talk about one method, there are, this is not the only way to do this but I am going to talk about one standard way to do this. And this can be employed anywhere. It could be employed whether you use Otsu or whether you use a different method like Chow-Kaneko or whatever, right? This is just a scheme which is some kind of an interpolation scheme and because of the fact, right, that you have very sort of, you know, you could have - I will explain what I actually mean by that, I mean so even when you actually split these regions like this, it could turn out that for some of these regions, you cannot even assign actually a threshold.

Now I will explain in more detail what I mean by that. So, when you split it into smaller regions, you might end up with situations where some regions you can't even assign a threshold. And therefore, right, what happens is if you wanted to do a thresholding then you will have to, you know, interpolate from your, say, neighbors. I mean you will have to guess a threshold for such regions by using the threshold values in it, say neighborhood.

And because that will involve sparse, right there are probably not kind of too many regions and, you know, it is not through that every region will have a threshold and therefore, this interpolation is kind of a little more tricky than the one, which we have seen earlier. For example, when you did the image rotation translation and all, right, we were doing a kind of binary interpolation.

But there, you know, there was always a guarantee that you had four neighbors that had values, there is such a guarantee. It does not even exist here and therefore you know; this requires a different line of thought. We will eventually also try to use binary interpolation when we finally want, you know a pixel level sort of a threshold. I will explain that.

But right now, as we stand, what we need is a thresholding scheme, an interpolation scheme that will assign a threshold for each region. Okay, now how does this work? Now this adaptive kind of, you know, this one, a thresholding method, so adaptive thresholding is used, is employed when a global threshold is not appropriate, a global threshold is not appropriate, is not appropriate. Okay, for example due to illumination gradient and so on.