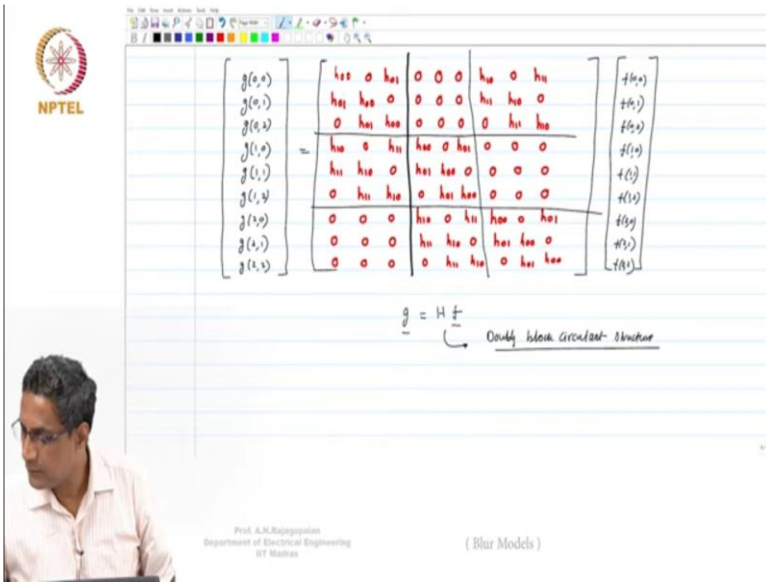
Image Signal Processing
Professor. A.N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture No. 24
Blur Models

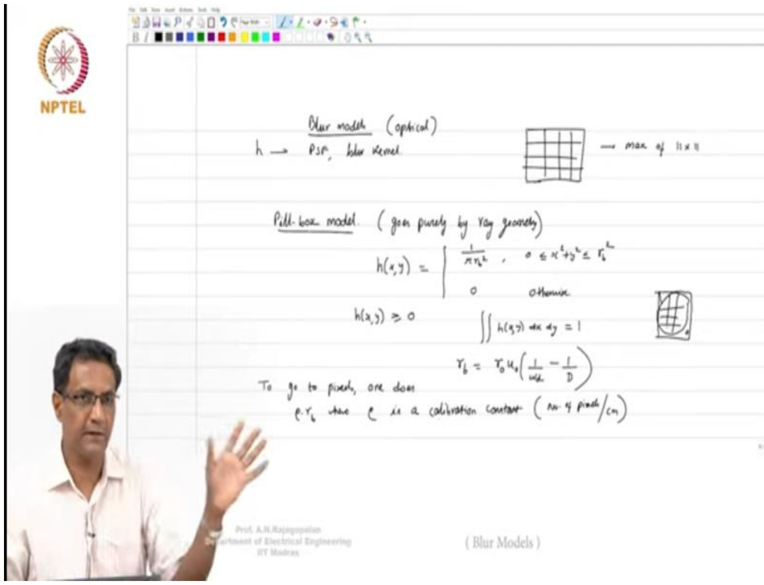
(Refer Slide Time: 00:17)



Now at this point of time, let us actually move on to some some models for this H. If you are thinking how big will H be, see typically I have taken I have taken this H to be of the same size as f. In this case just 2 or cross 2 or something but ideally what will happen is this H will be a very very small it will have a support that is very small that means it will be non 0 only over a few elements, like seldom will this H go beyond 7 cross 7 and all, that is the image will be like final 5 cross by 12.

So, this matrix will typically be lots of zeroes. It is not usually like this. It will have lots of zeroes and there are various algorithms that actually exploit the sparsity of that matrix and so on but then the circulant everything will still will still remain except that H is typically H will typically have a very small support and where this H comes from and how do we typically model is what is what is what we will talk about next. What is called blur models?

(Refer Slide Time: 01:14)



Blur models and it turns out that in only for optical blur you can have these models. The motion blur which is the other kind of blurry, you cannot have these kinds of models in all format. So, this is again only for the optical case, just to just to make it clear. We cannot take this take this and use it for motion blur and all. So, the blur models we and we also call this H rate is referred to as either a PSF or we call it a blur kernel. Kernel is just a just a commonly used term for for a blending function or a blurring operator. So, you can think of it as a discrete case. This will be a discrete discrete set of values.

This will be a 2D array with some discrete set of values sitting inside, with some with some with with some values sitting inside it except that on the spatial grid it is actually a discrete kernel. Now, so the size is typically max of so you will not see more than 11 cross 11 and all, very rarely, not that it cannot happen but typically we do not go, any image, if we take and blur it with kernel it and and the idea is that we do not go around blurring and all. It is simply to understand that that is what happens inside the lens. So, it is not like we are interested in taking a taking a good image and spoiling it. That is not the way. That is not what I mean.

What I mean is when you see a blurred picture, you should realize that there is some PSF like this that is acting on the original pin hole image that you should have ideally seen in order to be able to give give this kind of a blurry picture and that the blurriness itself if you see even with a 3 cross 3 or a 5 cross 5, you can see so much of blur that you do not need something very big and all for it and generally it is not.

Now, the first model that is commonly used is what is called a pill box model a bill box model, it is similar to a cylindrical, a pillbox a pillbox is what where you where you where you are going to keep the pills at home. That is a pillbox. Looks like a cylindrical thing. So, the pill box model, it goes it goes by ray geometry purely by ray geometry. Purely by ray geometry and according to which under the blur kernel, I am just writing a continuous case here. We will talk about how we how we normally do a discretization. So, continuous cases easy to explain.

So, it goes like 1 by 1 pi Rb square as long as x square plus y square lies within Rb square and 0 otherwise. So, what this means is that and again as you can as you can clearly see, one oneof the things that any kernel should satisfy, blur kernel should satisfy the fact that h xy should be greater than or equal to 0 and then double integral hxy dx dy should be equal to 1. This is something that we said already. So, any any PSF for that matter should satisfy this condition, if it is a point thread function and as you can clearly see here, it is 1 by rb square.

If you integrate that is the area under the so it will automatically integrate to 1. Outside of it, it is all 0. So, so it is like saying that inside the circle so you have got intensities and outside its all 0 but normally what we do is, we like to deal with kernels that are regular in shape and not like circle and all so what we will normally do is, we will kind of take, we will inscribe it within within a square and then outside the circle we will say this is all 0 and then use that as a kernel. That is what is more common.

Why does greater than or equal to 0 because I mean, you cannot really reduce it, you cannot make an intensity negative, through when a when a lens acts, a lens cannot make the intensity negative. I mean it can only spread? It is like saying that if I if I give give an intensity and if you see a blur, because no blur, you just see the see the intensity that you would have seen as a pinhole else it has to spread.

If you have a if you have negative weight, I mean, when a lens is not supposed to take away things from the energy it can only, it will not take anything, it will not add anything but whatever is there it should come up on the image plane. So, the so the so the integral should sum up to the original intensity.

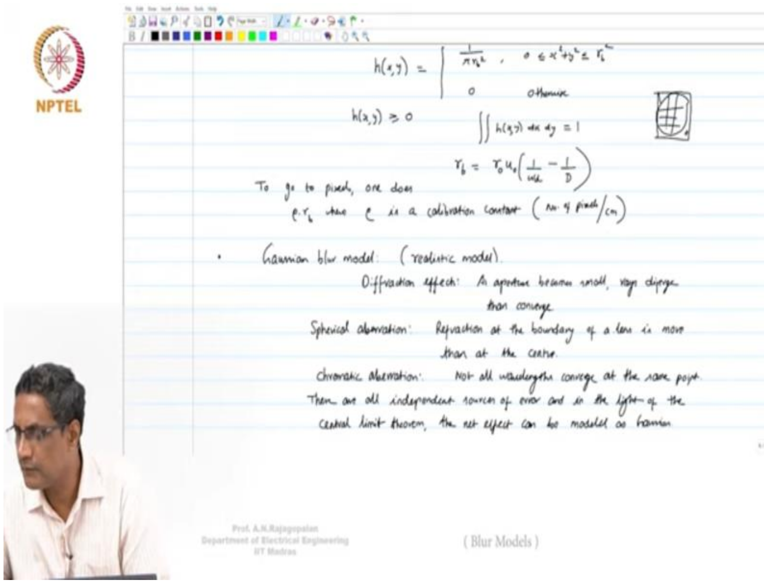
So, we think of that as a blurring operation that the blur kernel should be such that it sums to 1 and then all its weights are greater than or equal to 0 and so one of the things is this and so, so the thing is that it is almost like saying that you know outside this blur radius. This rb is the same blur radius that we talked about. So, this rb is is actually the blur radius that what was that equation r naught or t was it, did we use u naught or v naught. I think we used u naught maybe so r naught u naught into 1 by omega d minus 1 by d.

This was the expression that we are actually derived for a lens. So, this bar rb is the same kind of blur circle and and in order to express everything in terms of in terms of the pixels, to to go to to go to the pixels because you know, we do not talk about blur radius in terms of centimetres in law. What is normally done is, you multiply rb by rho. Rho times, 1 does rho times rb, where rho is a calibration constant, which is, it sort of tells you calibration constant whose unit is like number of excels for that sensor, for that particular lens, for that particular image plane that you have per centimetre.

So if you have an rb, let us say you have a sensor resolution such that you have got your pack like 5 pixels per centimetre and if you have a radius like 2 centimetres and it will mean that you have a blur kernel which is extending to a 10 pixel size because, eventually you have to have everything on a sort of a discrete grid.

So, this row is something that, let us say, one has to calibrate a priori and then one can continue to use it but this model this assumes that the rays are all it strictly limited to a circle outside of it and everything will just fall off to 0 immediately after that and so on. So, more realistic model is actually a Gaussian model which I will write next.

(Refer Slide Time: 8:00)



So, more realistic thing is actually a gaussian. So,` that was pillbox. The second is actually a gaussian blur model. This is what is more common but in MATLAB if you see, they will they will allow both, they will accommodate both but the gaussian is more common. So, here it means that all the all the intensities within the circle are all uniform. They are all equal.

So everywhere, it is like 1 by pie rb square and outside it just truncates off immediately to 0. The gaussian blur model makes use of the fact that this is a this is a more kind of a realistic realistic model, in the sense that what it says is the norm. It takes into account various. So, one is one is what is called what what is called a diffraction, a diffraction effect. So, every system is actually a diffraction limited.

Diffraction effects what that actually means is means is that as you kind of keep reducing the aperture size you may think that you may think that you will end up with a more and more sharp image, but then beyond a point when the aperture begins to become smaller than a point then the rays actually begin to diverge. That is that is actually a diffraction effect and these are actually real effects.

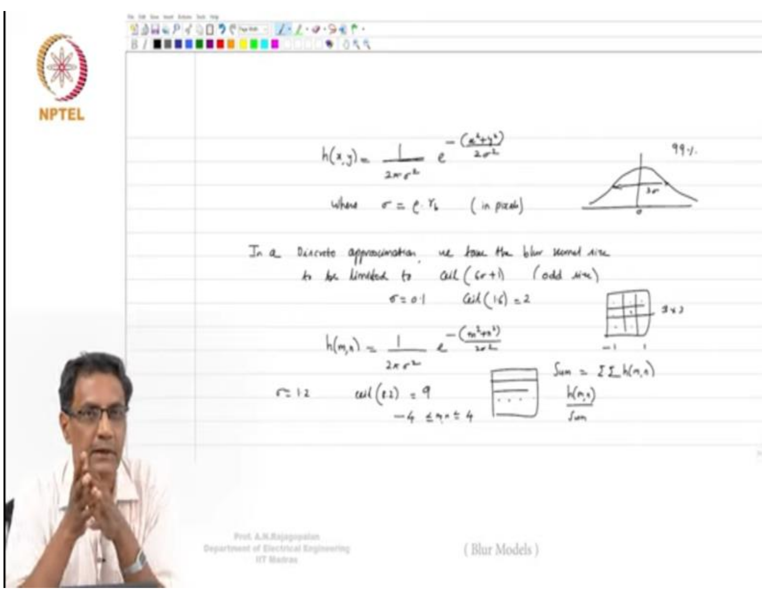
So, so so basically means that, so if I have to write it write it correctly, as aperture becomes small, as aperture becomes small, rays diverge begin to diverge then converge which actually means that this assumption that everything will simply be 0 outside the radius of rb and all may not actually be true because there will be something that is actually going away.

Defraction effect is one, then is something called spherical aberration of the lens. Spherical aberration what that means is refraction at the boundaries of a lens, refraction at the edges or boundaries, the boundary of a lens is more than at the centre and the and the other thing is all these are actually independent effects. Diffraction is independent of whatever happened then and there is something called a chromatic aberration called a chromatic aberration according to which not all wavelengths will converge not all wavelengths converge at the same point.

This may not be may not be directly; you may not be able to see it directly visualize it but but not all will converge exactly the same point. So, there could be a small, now we are assuming that all of them will converge and we will get exactly a point of zone. So, there could be done some spreading around that is going on and there are also the other effects. So, the main main effects are these and you know, and these are all independent effects. These effects, these are all independent sources of error in the sense. These are all independent sources of error to the actual model that you have in mind.

All dependent sources of error and in the light of the central limit theorem, the net effect of all these independent sources of errors can be modelled as a gaussian. The light of the central limit theorem, which you know, the net effect can be modelled as the gaussian. So, which is why in MATLAB and all, you will find that the gaussian, it is what, let us say people you know, they want to synthesize some blur and all, they will go use a gaussian.

(Refer Slide Time: 12:08)



If you are wondering why they use a gaussian, this is the reason and which then means that the so you are sort of saying that h of xy some 1 pi or is it 2 pi sigma square, e raised to minus of x square plus a square by 2 sigma square again. So, 1 by 2 so that it sums to 1 integrates to 1 and this sigma, what is a sigma now, where the sigma is given as rho times rb. The rho still has the same meaning and so sigma is now in pixels.

So, if you say that I have Sigma equal to 1 pixel or something, now, the point is determined. So, what is actually saying is that the intensity will kind of fall off, and it will also it will also not something that will simply limit itself to only rb radius. If you look at the look at the ray geometry model, it seemed to say that outside of rb rate you will see nothing, but in reality rate what this model is saying is that it will kind of die down slowly, it will not just stop abruptly end like that.

And and then the other thing that it is also saying is that the intensity or the weight for the kernel be maximum at the origin and then about that will spread and then intensity will also fall off and this so so you might actually think that those then maybe this h of x is going to be so big then it but but but we all know that if you have if you have a gaussian, suppose I take 1D gaussian, then we know that then you know that most of the area, almost 99 percent of the area under a gaussian is covered within 3 sigma on either side.

So, if you go like 3 sigma, it has a standard deviation of sigma and if you go plus minus 3 sigma neither side, we know that we would have is almost covered 99 percent of the area under a gaussian. So, what we normally do is we so so so when we do a discrete approximation, so finally write what you need to suppose you have to implement something like this, you will need to do a discrete approximation, so discrete approximation.

So in a discrete approximation, we we take the kernel size or we take the blur kernel size to be limited to kernel size to be limited to to ceil of 6 sigma plus 1 because they is also the origin wherever they meet about the mean so 3 sigma 3 sigma plus 1.

So ceil of 6 sigma plus 1, so for example, if I say that sigma is 0.1 then you will say like ceil of what is it 0.6 plus 1 so 1.6, so we will say it is actually 2 pixels but then one likes to have an odd odd sized window because it is even then where is a centre and so on. So, we normally we normally limit to ceil sigma plus 1 but then odd size.

So, what that actually means is that even if I get 2, I will still take a kernel that is going from minus 1 to 1 so that I get actually a 3 cross 3 and then and then because then it is clear to me as to what is the centre. Even size nobody will take because that will create unnecessary problems. So, you take an odd size kernel every time.

So if you are not getting an odd number make it odd, next next next maybe an odd number and then in this case will be 3 cross 3. The only thing that happens when you do a discrete approximation, you will do it let us say, h of m. One way to do it, the simplest way is 1 by 2 pi sigma square e raised to minus of m square plus n square by 2 sigma square and you should remember that the centre is here.

So, m equal to 0 n equal to 0 is here. So, when you say 3 cross 3, you are going like minus 1 to 1 with 0 comma 0 at the centre. So, it is like I have a I have a blur kernel, so in the centre where my central ray comes, I will have the maximum weight and then around it will start to die off.

So, what will happen is if you just compute these values and if you put them here and they will not sum up to 1 because now it is actually a discretization. So, what is normally done is you compute for a sigma, you compute h mn, you fill all those values here then you simply compute the sum of say h mn calculated like this. The sum of all of them, they need not sum to 1 and then you scale h mm by sum. Each of the h mn. So, now that you know that it will sum up to 1 because you have to make sure that that that it is a blur kernel and it should not have something that is greater than 1 or less than 1 or something.

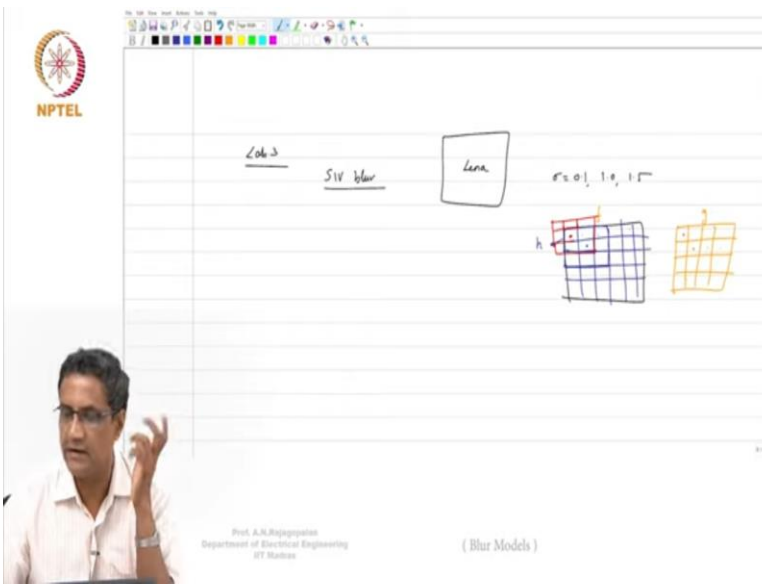
So, a simple implementation is where is m and n will go from depending upon suppose let us say if your sigma is the 1. 1.2 or something, then you will get like ceil of. What is that? So, 6 sigma 7.2 plus 1. So it is 8.2 so we will take 9. So, 9 is your total size. So, your n and n will go from minus 4 and so m gamma n will go from minus 4 to 4 and then for those values we will compute h mn. Plug it all in here and then take the sum of these values over this over this 9 cross 9 window.

So, it will be like a kernel. So, this kernel will be like 9 cross 9, 9 rows 9 columns, everywhere you will have 8. Add all them all of them up, you will get a sum divided every weight by that sum. So, now we have a kernel that that is a that is that is a approximation of this gaussian.

Now, at this point what happens is actually you people will have a lab for this which is only to only to get you to get you to get you time familiarize yourself with how do you actually how do you synthesize this kind of a blur? But in real real life, we may not want to do it at all. This is only to understand image formation process in a lens.

So, so by implementing it you will realize that is how the image is getting formed. Not that we want a chord that we will use to blur an image. Sometimes of course people use it. If they want to mask somebody’s face, they do not want somebody's identity to come up, to show up they do this but that is not our goal. This is only to understand the image formation process.

(Refer Slide Time: 18:26)



So, the first part of the (())(18:26) lab 3 and after this I will give a pause because after this I want to introduce mini projects for you and we will shave off a few of those assignments that we normally give and give them is optional. Now this lab 3. So, the way this works is the first is the space invariant blur. So, we will give you an image. Let us say it is some Lena or something. Now because it is all synthetic is there is there is nothing like a 3D Lena phase will this introduces blur or so on. We will just take some image and we will just blur it.

We will not, it is not like, it is not really reflective of what will happen when you if you were to actually take a lens and write image Lena. Lena’s face is like, it has some features and therefore there may be space variant blurring and so on but it is only synthesis in the lab. You can take any image that you want. You do not care about what its reading structure is. Just take that image and suppose I want to apply a uniform blur on it I will apply it. This is only it is only for that purpose.

So, we cannot start interpreting that if Lena had to be if in front of a lens then would it this is how it would have come. No. So, this is only only only only read for you to just play around with. So, you take a Lena image whatever image I do not know what we give. So, Lena and then we will ask you to blur it with sigma equal to 0.1 and if sigma is such that your size is only 1 in the sense that if I give you sigma to 0.01, then 6 sigma plus 1, all of that will simply end up with just 1. So, that means it is like an impulse. So, the blur is so small that there is that there is actually actually you do not see any blur at all.

So, the minimum size will be kind of 1, the maximum can go depending upon whatever sigma. So, we will, I think we start with even something like nobler which is be like 0.01 or something and then then then we will give like 1.5, 1, 1.5 and stop somewhere there because at that point itself else you will realize that the image has become very very blur and the way you will implement the convolution is exactly the same way that you would do in 1D.

So, here is your here is you're here is your Lena's image, and they are all intensely sitting now in each of these boxes and you will take. Given the sigma you will actually, you will you will have the kernel, you will actually build the kernel assume that it is 3 cross 3. Then what will happen so you kind of place that 3 cross 3 kernel here. So, here will be the centre, here is where you want to know what should be the value of G now so you do the simpler and so now you can slip the colonel but this is all isotropic, it is gaussian whether you flip or not does not matter.

So, if you want you flip, if you want a proper code flip it both along X and Y and then what will you do, you will overlap it with the with the image under it and then the image has some values here under you're your h has some values in it multiply both and then add all of them and then the then copy them into into G. Do not copy it back here. You should not copy it back here. You should have a g.

So this is f, then f is the black one, h is the h is the orange, whatever brown one and then you simply copy these intensities. Now if you are wondering what do I do with the boundary and so on so so so this value that you get after convolution, copy it here then slide to the to the next location, do the same thing find out the convolution. That is how you will get a picture that look blurred. Fairly straightforward.

This is this is straight forward. Now, I just wanted to tell you that at the boundary because what will happen, you also have to find the value here. So, you are sitting here. You have your 3 cross kernel here. You want to you want to say what value should I copy here. So, only thing is, here the problem is outside you do not have any values at all. So, but in order to make sure that you will not end up reducing the intensity just because there is nothing outside what we normally do is, we ask you to just look at only at the boundary.

Take these 4 values under the kernel, if it is 3 cross 3 and then scale them by their sum so that they sum to 1 and then simply take the intensities along and copy that here. Just take a weighted average like that or you can make an approximation that this that this image itself repeats left to right. So, this column wherever you can kind of think of it as folding around then you can do a whatever approximation you want or we do not really care even if we just copy this intensity there and put it we do not care because this is only at the boundary.

So, people make different different approximations. You can make whatever approximation you wanted at the boundary.