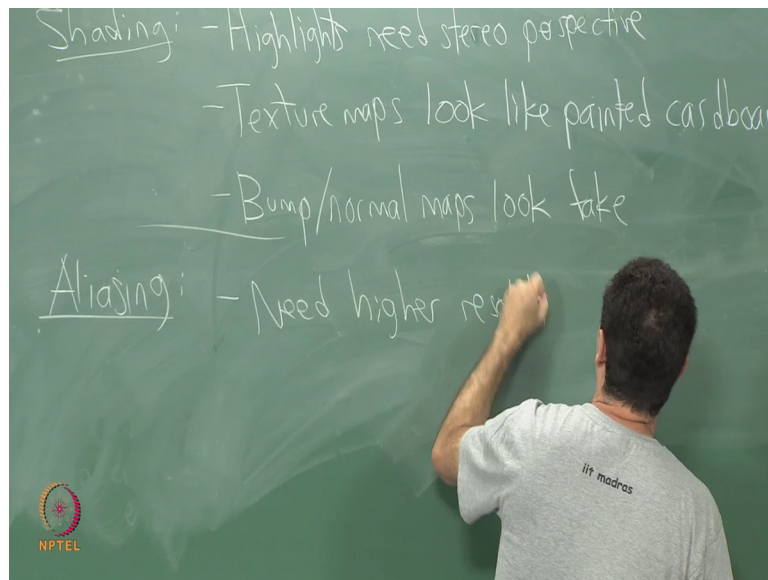


Virtual Reality
Prof. Steve Lavallo
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture – 16-1
Visual Rendering (VR-specific problems)

I want to now summarize some of the problems with rendering for virtual reality and this may be one of the most interesting parts of the lecture then for you. Mainly it is been covering standard techniques from computer graphics which are used for rendering to a screen that is fixed in space where the viewpoint is fixed typically.

(Refer Slide Time: 00:46)

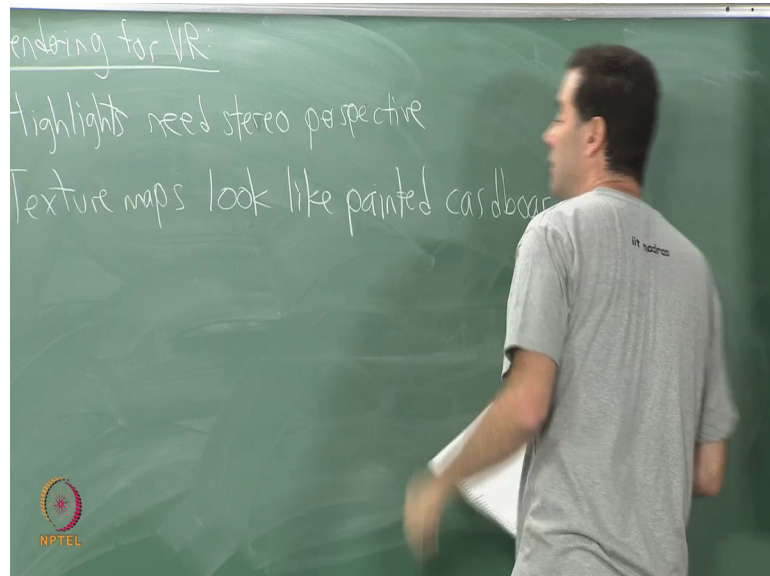


So, let us look at problems with rendering for VR ok. So, let us go into the shading part alright I gave various shading models and one of the troubles is that the highlights or specularities need stereo perspective. So, you have to do twice the works there right. So, you have to take into account the left eye and right eye perspective I mentioned this already; if you do not do this it would not look right you know just the convergence will not look correct on the on the surface of the object. Texture maps I just erased texture maps if I do texture mapping on a screen you know to some to some surface that is rendered on a screen and your in the world looking at the screen and may seem fine.

If you do this in virtual reality you can put your face up to the surface and you start moving your head about and you look at it with your eyes and stereo and basically text

your maps look a lot like painted cardboard. So, so you cannot really; you do not really perceive the texture with all of its depth it just looks like a painting which is what would happen in the real world right.

(Refer Slide Time: 02:35)



If I were to just paint a carpet like texture onto the surface of this desk here I could count and say it looks like its painted it does not really look like a carpet it is much harder to see that on a fixed screen, but in virtual reality is very easy to see that. So, it does not look right.

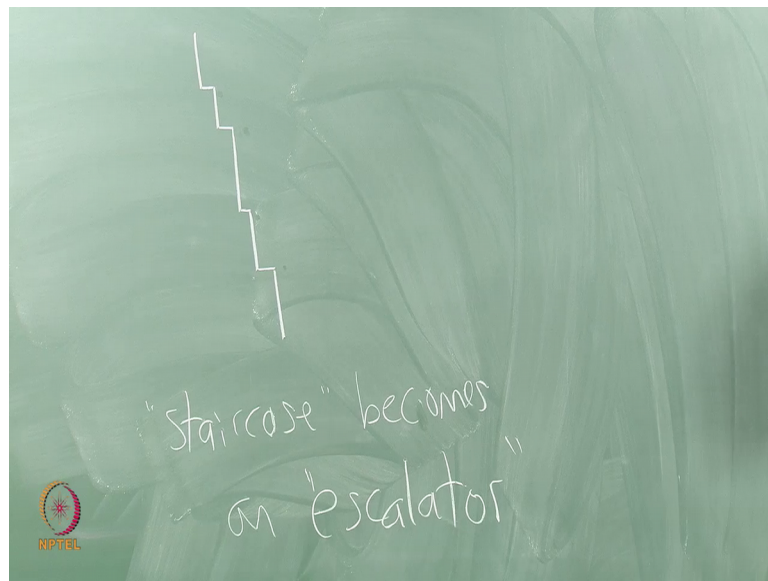
So, texture maps look like painted cardboard our painted walls. So, you can tell the textures artificial because you can change your viewpoint you have stereo looks artificial. Bump maps and normal maps look fake; you can very easily change your perspective you come up you look at it say no that is wrong. So, it is very easy because you can easily change your viewpoint it is easy to determine that these are not done correctly.

So, those are some challenges there is here is some well known tricks and these in virtual reality they do not work so, well. So, what should we do that is for you to figure out let us go to another problem aliasing excuse me. So, we talked about aliasing this is the problem of not having enough resolution and we have already talked before about number of photoreceptors that we have on the retina especially at the fovea. We talked about putting the display in front of the eye and the problem of seeing the pixels right.

So, it seems like we need higher resolution displays than ever before. So, aliasing is clearly going to be a problem.

So, we need higher resolutions than before just because of the way that these the optical system is arranged. So, we need that and I think I am going to go over to the other board here getting too low ordinarily.

(Refer Slide Time: 04:47)



When you look at a surface and in a rendered surface on to a display with pixels if there is not enough resolution you get some kind of staircase effect on the boundary right.

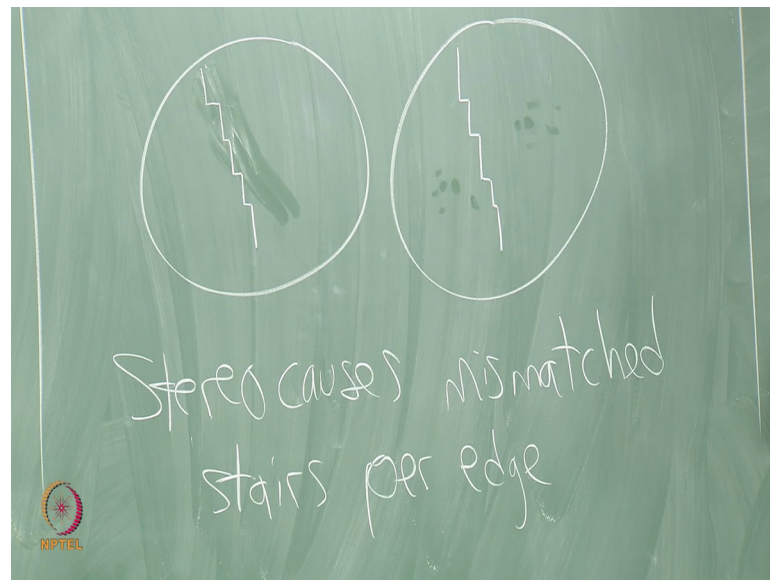
So, something we are quite familiar with if what is being rendered here depends on how you looking right which way your head is moving; remember we have motion tracking occurring that was a couple of lectures ago putting all this together we are tracking your viewpoint; that means, that the particular stairs here if you call this a staircase effect right this looks like a.

So, this is called a staircase effect the particular stairs will look like they are moving up and down. If you are moving your head right even the smallest motions that you do will be very easy to see here as what looks like some kind of rippling or changing of the stairs and because our eyes are very sensitive to motion we tend to focus on that.

So, we notice these flaws very well. So, not only is everything somewhat zoomed in let us say to where we know we notice pixels, but in addition to that we see these jagged

boundaries moving around. So, so, what I like to say is that the staircase becomes an escalator. So, the fixed staircase that we see on a screen becomes like moving stairs and your eyes fixate on that; now it gets even worse because we are really looking at this in stereo.

(Refer Slide Time: 06:20)



So; that means, I have I have two different views one for the left eye one for the right eye and in one eye view I guess I should make the staircases really look right here I guess.

So, so I have one view for this eye and have another view here for the other eye right and the particular stairs do not match up exactly do they because it is a different perspective for each eye. So, now only do I have an escalator I have multiple escalators I have two different escalators one for each eye. So, they are moving up and down and I am trying to converge on a single light.

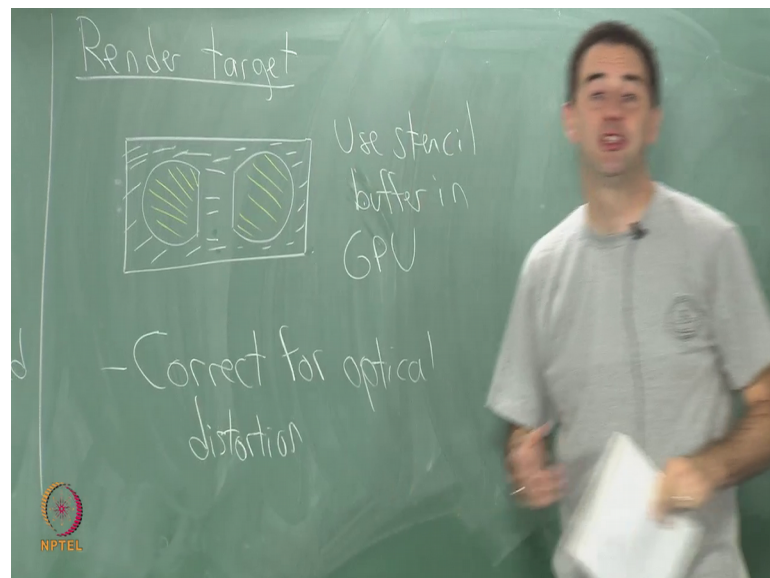
So, in the real world there is no escalator I converge my eyes together and I look at a perfect edge here I have moving escalators on both eyes and they do not match. So, I see this fuzzy sparkly kind of escalator pattern every time I looks at a sharp edge. So, see if you can find that in some of the in some of the demos that you are trying. So, stereo causes mismatched stairs or mismatched escalators for every edge. Generally speaking the higher the amount of contrast between this side of the edge and that side of the edge, the more you see this effect right.

So, if there is a very bright. So, if you have black on one side white on the other you will notice this. So, if you want to try to make a display or a project where you can highlight these effects then try to put black on white make it bright against dark and you will see these effects come out very easily.

So, aliasing is really bad because of these questions about that; sorry I do not have the solution. So, you have to, but they are just pointing out the problems part of our jobs as researchers right is to find the problems. And I think we are at it we are at a place in this field where there are many more problems to solve than we know what to do with.

So, there is a lot of good activities for you as students and beginning researchers.

(Refer Slide Time: 08:44)



Another problem is the render target normally the rendering is being sent to a rectangular screen. Now that might still be the case if the rectangular screen is put inside of a head mounted display, but due to the way the optical system works only some part of that is actually visible. So, even though there might be a rectangular screen inside of your head mounted display although very soon these displays are going to be optimized.

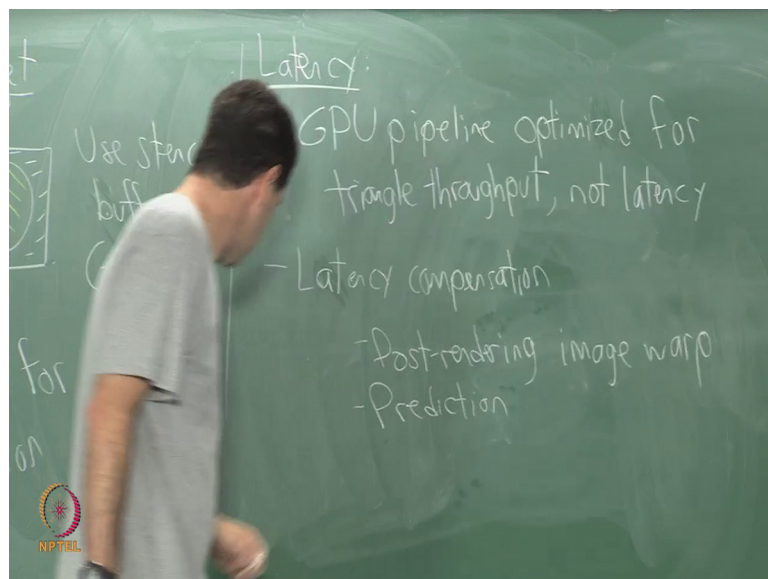
So, that perhaps there is one display for each eye and it may be shaped appropriate leave and it may not be a perfect square things may be optimized in some way, but as it stands right now; if you render to a display there is maybe one region that you can see for one eye. And another region that you can see for the other eye and the rest of it out here you

might as well not even render anything to because you cannot even see it. So; that means, the effective area which I will just shade in yellow here this is the place where you need to be rendering, some kind of shape like that anything you do out here is just wasted work.

So, that is a little bit of a difference not too hard to fix it turns out if you just use standard graphical processing unit GPU tricks you can use what is called the stencil buffer in the GPU to just deactivate all of the pixels that are outside of the actual field of view due to the optical system. So, you can just turn those pixels off using the stencil buffer setting the right flags and do that in one pass. And then it will quickly skip over that the GPU will quickly skip over those pixels when you when you do your rendering.

So, that is nice the next problem you have to account for which I have talked about before and I am going to talk about it even more in just a little bit is you have to correct for optical distortion. This could correspond to these spherical aberrations of the lens could correspond to chromatic aberrations or other kinds of aberrations you have to correct for the optical distortion of the lenses which I have not talked about yet in the context of rendering right. I have mentioned it before in previous lectures questions about that?.

(Refer Slide Time: 11:17)



Two more problems latency; so, one fundamental issue is that GPUs. So, whole GPU pipeline of computation has been optimized over the years for triangle throughput. In

other words blasting as many triangles per second through the pipeline as possible not latency. So, if you waste a frame or two doing some kind of pre computation.

So, that you can get as many polygons as possible through the pipeline no problem right, but now if you waste a frame or two in virtual reality that introduces latency in to the system that causes discomfort and possibly simulator sickness. So, GPU is need to be re engineered to optimize for latency not necessarily blasting as many triangles as possible, until you fix the latency problem you know if you get the latency down enough. So, that you can go through it maybe 90 or 100 frames a second and there is no latency in the graphics pipeline beyond that you can render everything you need in one frame time will be in very good shape.

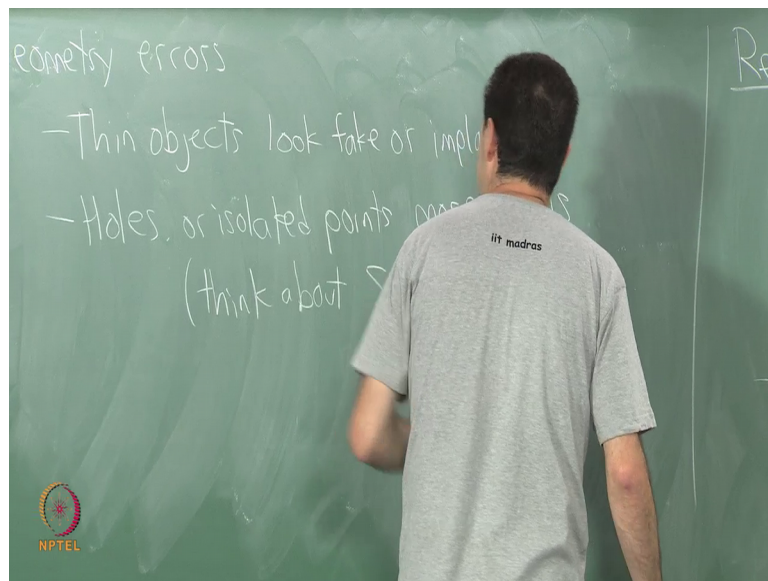
And then only then I would say try to optimize the number of triangles that you can blast through the pipeline if your geometry needs to be simpler because of that. So, be it will be a comfortable experience you do not need to have absolute photorealism to have a good VR experience, but you do need to have the latency brought down as much as possible. So, I am going to talk about some latency compensation methods in just a little bit. So, you can do latency compensation, but the best way to do it is just to remove the latency in the whole rendering pipeline; there is not very much latency in the tracking part its mostly in the rendering part right.

Now, whatever a little bit there is in the tracking part can be easily compensated for with prediction, but based on the momentum of your head and a little bit of prediction, but the comp in the latency part in the in the graphical rendering pipeline is large enough to wear for a very complicated scene people right now are doing some kind of artificial tricks. One of them is post rendering post rendering image warp which I will talk about shortly and you could also do additional or prediction using your tracker.

If you do not have to predict too far into the future, it will only compensate for part of the problem. And if you know precisely when you are rendering is going to be finished and you are actually going to be switching pixels. If you know when that time is going to be you could do some kind of prediction using your tracking algorithms, but the further you try to predict into the future the more you might introduce errors or instability into the into the system which are perceptible and that may be a problem.

So, here is a couple of ways to help with this while we operate or we live in an era of high latency hopefully we can eventually have GPUs that are well optimized for latency and not worrying as much about blasting or maximizing the number of triangles blasting as many triangles through the pipeline as possible alright. One more topic and then I will get into some of the ways to mitigate these problems, but certainly not there across the boards.

(Refer Slide Time: 15:08)



So, there are in the models that we use a lot of geometry errors geometry or geometric errors as I say geometry errors.

So, thin objects it may be the case that for example, the geometric model might have some leaves on a tree. And these leaves are very thin you know leaves in the real world look very thin, but they may be. So, thin that they generally have a thickness that is much lower than the width of what a pixel would typically be right I mean any kind of rendering that we do so; that means, that we want to make very thin objects and if you render them on a screen they might look fine.

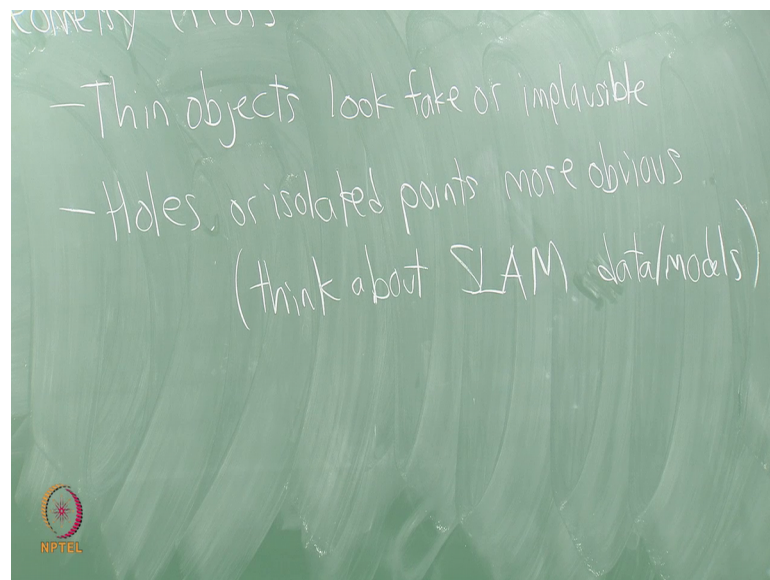
But if you put your face up to let us say a leaf and you look right down its edge; it will look incorrect it will not render properly it looks very fake. So, I am; so, thin objects look fake or implausible when you combine that with aliasing problems it looks even worse, try that out in some of the demos that you have on the lab take a look put your face up really close to leaves and start looking at them from different perspectives let me

know if they look right they will also be texture mapped in many cases and then it will look like as if the leaf texture is just painted onto some surface that is way too thin.

So, things just do not look right what if you have holes in your surface. So, holes or isolated points maybe there is just a point raised above the surface become more obvious. In other words whatever minor flaws you have in some surface now if you rendered this you know with a cad tool or maybe you wrote some code that generated a perfect surface maybe it is going to be fine.

What if you got the data using some kind of simultaneous localization and mapping method right you have some kind of a depth cameras you are building some kind of view you reconstructed this table that is in front of me for example, it might look fine when you render it on a screen and you are just sitting in your office looking at it, but if you go put your face up to the model and you look you say oh look there is no data there oh I see a hole there is a mess there it is not done correctly. So, once you can move around through the space the data looks a lot worse.

(Refer Slide Time: 17:49)



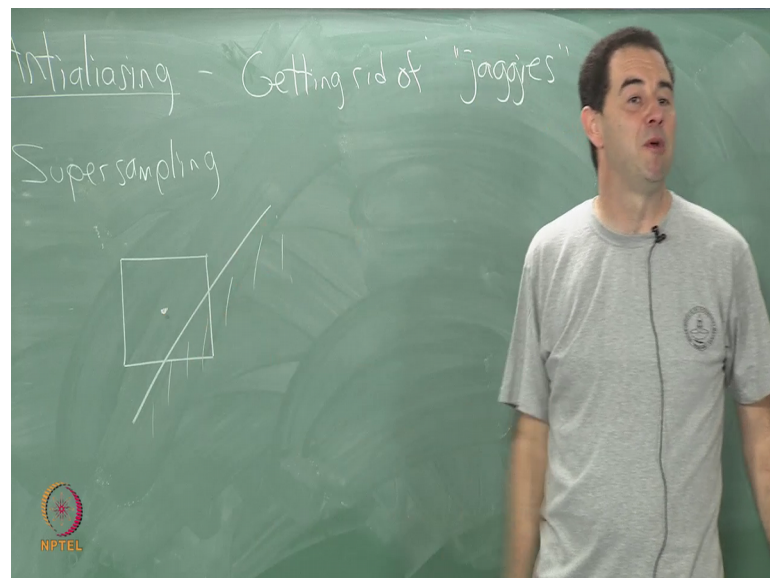
So, any kind of model that you might have generated automatically right may look worse. So, think about some kind of models that you might have generated from slam or simultaneous localization and mapping slam data or models alright; questions about that? And finally, you know one more bit about the geometry remember we have talked about this before what do you do if you come up to a wall and you put your face up to it.

In fact, what if you want to just want to in virtual reality just go right into the board in the in the real world my head hits the board and I am stopped was not completely comfortable, but it does not nauseating in the rit in the virtual world what happens when I come to the board do you want to just artificially stop the head motion that it could cause a vestibular mismatch that is nauseating.

Or you can poke your head completely through do you really want people doing that in your virtual world right maybe they poke their head all the way through and they start talking to people in the corridor outside alright that is not right it I s not what you intended. So, either make them sick or you give them the ability to poke their head out into the corridor which one do you want right or you can just make it.

So, if they cannot ever stand close enough to the board. So, they cannot get there then they get frustrated. So, you have to deal with these strange problems that people do not spend a lot of time talking about, but this is what happens when you try to build a VR experience alright. I am going to talk about a couple of I got some talk about three different kinds of techniques to handle some of these issues, but they do not get us all the way there. So, I just want to talk about some of the stay weird techniques.

(Refer Slide Time: 19:33)



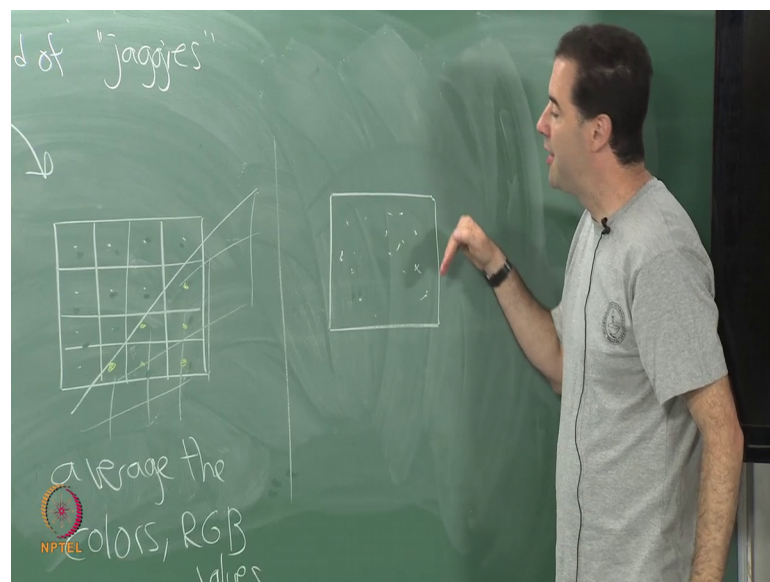
First I will talk about antialiasing. So, as I said we have these moving staircases or escalators one per eye and ends up being a big mess. So, antialiasing is now more important than ever maybe there was some hope that aliasing issues were not a problem

as the resolution of displays got higher and we can barely even see pixels anymore any of our displays in the real world now in virtual reality we have problems with pixels again.

So, antialiasing whereas, people in the graphics industry like to say are getting rid of the jaggies back in the 1980s especially people call this aliasing problems the jaggies jagged is what it comes from jagged lines. So, you have a bad case of the jaggies you need to fix it. So, the most common method is super sampling which is you know in the standard method we have a pixel; I will just zoom in on the pixel and I have my triangular edge hitting here. So, this is overlap that I have and remember I said we just sample at one place and determine is that does that fall inside or outside of the triangle in this case it is outside. So, we do not render that pixel according to this surface of the triangle.

Here if this point had fallen on the inside then we render it. So, it is a binary decision that is all we have inside or outside. So, another way to handle that by super sampling is we divide this pixel into sub pixels.

(Refer Slide Time: 21:04)



I am not doing a very good job here making it uniform well let us pretend I drew that in a uniform way and then we just sample each one of these. So, we pretend we have a much higher resolution and we sample points inside of here and again I have the edge slicing across in some way.

So, in this particular case I may have found that these points fall on the inside of the surface here this is the triangular surface. And so, these points these 1, 2, 3, 4, 5, 6 are on the inside these 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 are on the outside. So,; so, I can say that six out of sixteen points are on this side, so then I will do a kind of interpolation or I will average the values right.

So, I will figure out maybe these points correspond to some other surface right behind that may be another triangle or in front there; however, it works let us say behind maybe another triangle because this is occluding some other maybe surface in the back. So, I will do some kind of averaging between the between whatever properties. So, say RGB values that I would have for the surface behind and the surface in the foreground and I will make that ratio be 10 16s for the background and 6 16s for the front.

So, I just do an interpolation. So, that is a way to assign this a kind of averaged value that will in some way smooth out the jaggies or the aliasing; rather than making it all one surface or all the other surface which only amplifies this problem at the boundary. So, that is one way to fix that. So, you average the colors or RGB values or whatever else they are going to be using as attributes you do not have to pick a regular pattern like that. In fact, that may cause trouble in a lot of cases.

So, you could pick some kind of random pattern of samples right and then I am again figure out what fraction falls inside, what fraction falls outside just from your samples you could use random samples or you could use some kind of irregular deterministic sampling pattern such as a low discrepancy sample sequence. So, all sorts of ways to do this what is a disadvantage of this? Now it looks like you have to essentially do these computations these shading computations at a much higher resolution than your image right.

So, now, it is very expensive; so, this may help with the aliasing problem, but you now had to draw images as if you had a display that was much higher resolution and so, if you already had trouble as latency then fixing the aliasing problem may give you even more latency. So, that is an unfortunate disadvantage of this is that it is relatively expensive.

So, good, but expensive one way around this which is done in GPUs is to use what is called MSAA multi sample antialiasing and in this case this higher resolution sampling is only occurring for some of the attributes. Such as depth or whether not you are on one

side of or other of the triangle, but the more advanced shading calculations are not done for all the samples and they try to put together a result that looks good enough and people accept that.

So, there is some kind of let us say a hack that is done in GPUs using MSAA you can read about that if you like the more details of it as various techniques for that and it seems to provide reasonably good performance does it still look good enough for virtual reality; I do not know you will have to do your own experiments and see, but this is some kind of let us say trade off that is done in GPU. So, that you do not have to fully do all of the rendering calculations at a higher resolution.

But just do some of them and hope for the best questions about that? So, when you say a little bit about how aliasing problems are handled, but it comes at a great expense or there is a kind of trade off that seems to give some of the benefits at a lower cost.

Next problem is handling the distortion let me just cover the distortion due to spherical lens aberrations.