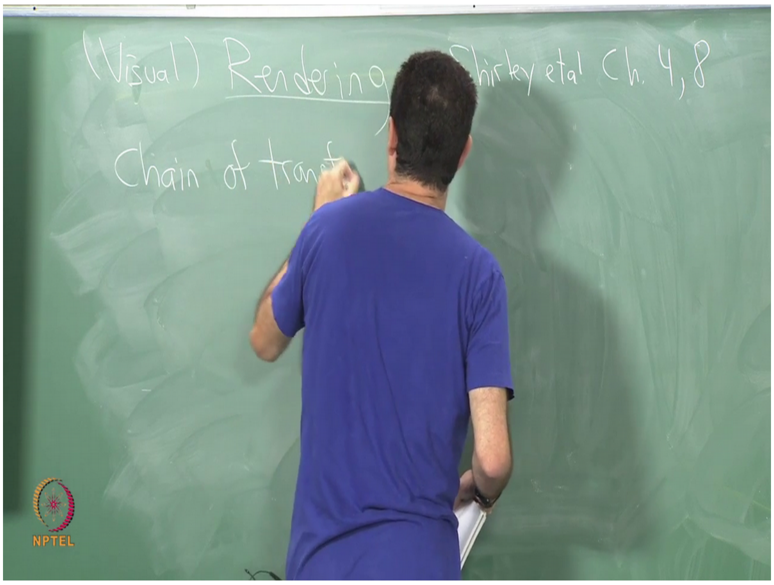
Virtual Reality
Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture - 14 - 3
Visual Rendering – (overview)

I am mainly going to get into this in the next lectures, but let me just give a little bit of a introduction today.

(Refer Slide Time: 00:21)



Rendering and I mean the visual part because, you might remember that early on in the course, I talked about rendering and displays for other senses right, it could be for audio could be for haptic or touch. So, in this case we are talking about the most complex part in the most common part for virtual reality which is visual rendering, if you are following along in the in the textbook, I suggest looking at surely at all chapters 4 and 8 before the next lecture.

(Refer Slide Time: 01:04)

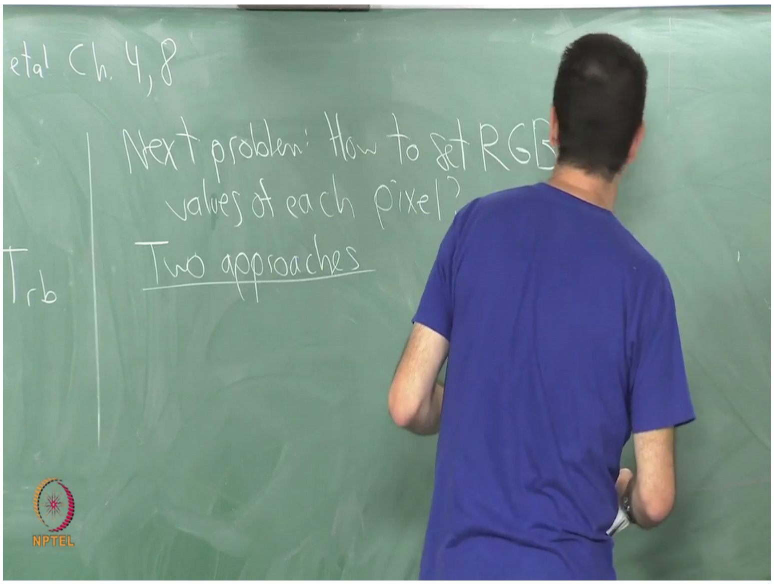
So, you might recall we did this thing called the chain of transformations, and that was so, that we could transform a point on a body into the world, and then eventually into the appropriate place in pixel coordinates right. So, we went through that whole chain of transformations, we finished without a few lectures ago, we ended up with something that looked like this, we had the overall transformation is equal to T viewport, T canonical view, T I and T rigid body, and I went through all these transformations in detail I even insert it for stereo T left and T right in here.

Or I might have combined them into making a right eye and left eye transformations here, by replacing this matrix, I also talked about optical distortion transformation out here, but what was important about the previous transformations was that I was just telling you how the geometry get us transformed. So, we could figure out what the coordinates are starting from my initial objects, that I define in their own body frames, and I put them all together in the same world frame, I just want to figure out where everything ends up when it needs to be rendered on the screen, but I did not talk about how the pixels are actually supposed to be set.

So, now I want to get to that part I want to talk about how should the pixels to be set what information needs to be sent out to the display remember, we also talked about display issues when we were talking about eye movements, and we were talking about perception of motion and, we went in discussions of frame rates and resolutions photoreceptor density all these things are coming together now.

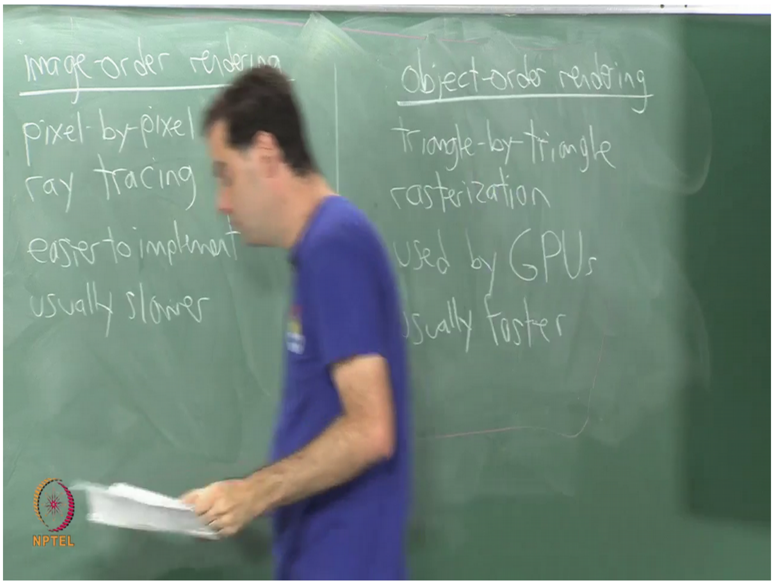
So, when I cover visual rendering I am mainly going to be covering for you what our standard computer graphics techniques that have been known for decades; however, we should look at them from the perspective of virtual reality developers and, how these methods may in some cases be adequate for what we want to achieve in virtual reality and, how in other cases they may be quite inadequate. And if they are inadequate what do we need to do about it right. So, that is the next problem this was pure geometry, under the mathematics of transformations.

(Refer Slide Time: 03:25).



So, the next part is next problem, how to set the RGB values of each pixel, there are 2 general approaches that emerge, again this is a standard computer graphics there is what is called image order rendering put it here, image order rendering versus object oriented rendering sorry object order rendering, I said object oriented because, we think like that in computer science, but object order rendering.

(Refer Slide Time: 04:11)



So, an image order rendering, and I should say to make the distinction between the 2 of these you have to think about what the outermost loop is going to do in the computations. So, the outermost loop in image order rendering is to go pixel by pixel. So, it is for all pixels and, then inside of that I figure out how to color that particular pixel by doing a bunch of work. So, this is going to be pixel by pixel the main method inside of this is called ray tracing.

So, you figure out what ray of light in the in a in a virtual camera imaging model that pixel corresponds to and, then you do a bunch of computations about how all the light would be propagating from fictitious sources, in order to eventually enter arrive at that pixel. So, ray tracing this is I would say easier to implement and, if those of you who have had a computer graphics courses before you may have had to implement a ray tracer as a class project. So, it is a very reasonable kind of assignment overall though the performance is usually slower.

So, nice from an academic standpoint and in terms of easy to learn and implement, but often ends up being slower, object oriented which the particular objects that w are going to use in this class are triangles just the basic primitives, when we talked about models I mentioned triangles is the most common primitives. So, this will be triangle by triangle order. So, in other words I start with thinking about my model of the geometry, and I just go triangle by triangle it is composed of triangle.

So, I go in that way that is the outermost loop instead of pixel by pixel. So, I focus on each triangle and, then I do what is called rasterization which is figuring out what pixels are covered by that triangle right. So, after these transformations have been applied right these transformations have been applied to the triangle from the original body frame that was given in or it may have been given in the world frame if it were stationary, where does it end up on the screen we perform some kind of pixelation of it in a process called rasterization.

This is much harder to implement what is what is used by modern GPUs graphical processing units, so at the whole graphics pipeline has been optimized for triangle by triangle computations quick rasterization, and fast operations by these processing units. And so, this ends up usually being faster, and so, what I want to do for next time is to give you an overview of how these kinds of methods work.

And then we will get into a kind of critique of the the problems that arise in the context of virtual reality, there may have been some methods or assumptions that worked fine for rendering graphics onto a screen when you are just sitting looking at the screen now. So, all of this was designed for primarily, but when we just adapted over to virtuality I want you to have an understanding of what may go wrong. So, are there any questions about this up to now? So, let us finish for this time thanks.