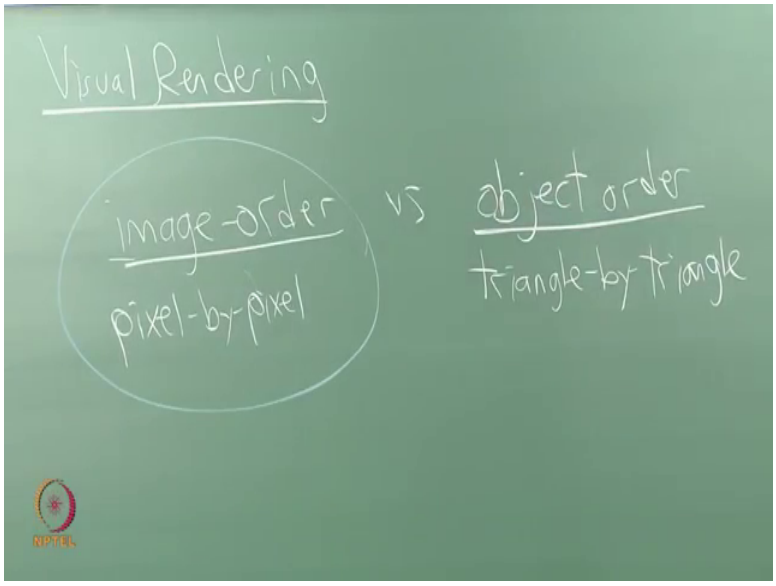
Virtual Reality
Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture – 15
Visual Rendering (overview, cont'd)

Welcome back. Today, we are mainly gonna be talking about the visual rendering part of virtual reality. I will be covering, standard computer graphics techniques, but, at the same time highlighting, what some of the challenges and differences are with respect to, rendering to head mounted displays and what the troubles are that are presented in that context, that are I think somewhat unique and unusual in comparison to graphical rendering on a screen, that is, that is fixed with respect to the observer. So, you may remember that last time. We covered tracking systems.

So, we finished up by covering both orientation and position tracking and. So, remember that the viewpoint that you have, when you are rendering the scene the, the virtual world or perhaps it is some recorded version of the real world, whenever you are rendering it for the eye, your eye's perspective is attached to a tracker. So, that perspective is changing frequently and that is one of the challenges that comes into play here, when we get into the rendering part. So, when I got into rendering I started to give a brief overview.

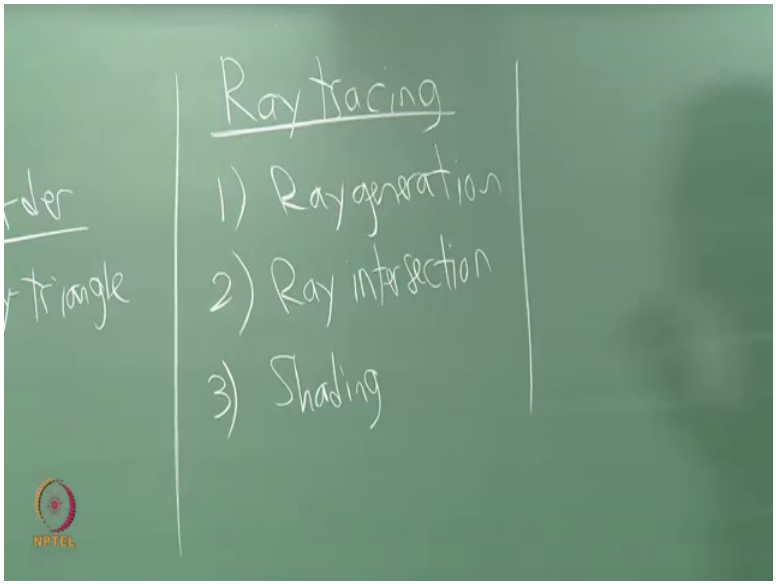
(Refer Slide Time: 01:33)



I will put visual in front of it, because we will also cover audio rendering and. So, you can imagine rendering onto a display for, other senses of the body and as I mentioned, there were two main approaches to visual rendering. there is image order versus object order, which an image order, it goes pixel by pixel and an object order. It goes object by object, which in our case is triangle by triangle. So, that is where I left off last time, let me just go through, what happens in image order rendering first.

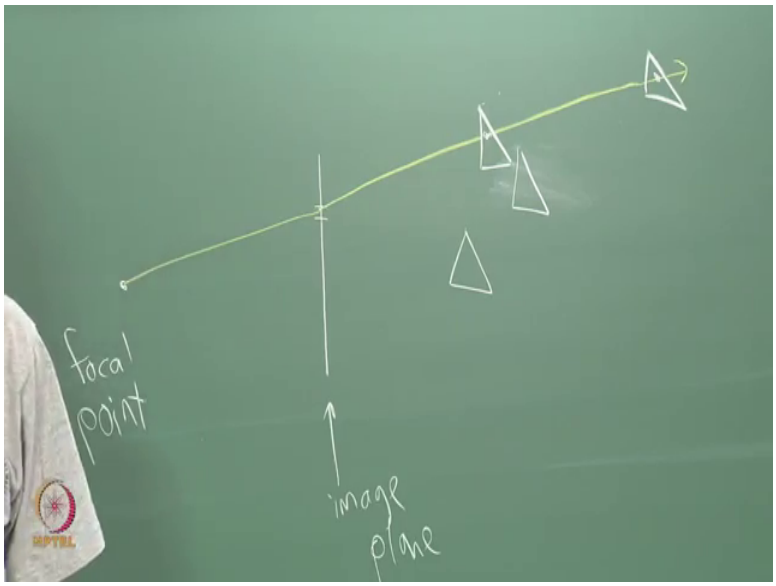
So, I will cover this case, but the particular methods inside are related to both. So, some of the things I will cover, end up applying as well to object order rendering. So, we will come back to those . So, generally the most common method for image order rendering is referred to as ray tracing, which involves, three stages.

(Refer Slide Time: 03:00)



One ray generation, two, I will write them here, quickly and then I will go into details on each of them, ray intersection and three shading. So, if we take a look at ray generation. So, it is pixel by pixel as I said last time, the outer loop in the image order rendering is to go across each pixel and figure out what should the R G B values be at that pixel. So, the first step is to figure out, what is the visibility ray that goes through that pixel to the focal point of this kind of a virtual camera, that we are constructing with perspective transformation. So, I will just draw it. So, that it looks like a one dimensional image here.

(Refer Slide Time: 04:01)



So, I have, have the image plane here and I focus on one particular pixel say, just has one location like that, there is the focal point of the perspective projection transform and then I have this ray that I need to define the equations of, I am not going to work through the algebra in this lecture. So, it should be a straight line and it goes out indefinitely. So, that is the ray generation part is to figure out, based on the pixel, based on the focal point. What is the equation for this, line, ray?

Let us say, which you starts with a line equation and then you just take half of the line in a particular direction going out and now, the next problem is to figure out what is the first one of your objects, which in our case is triangles to be hit by this ray going outward and. So, out in the scene there are some triangles around. Some of them are not hit by the ray oops. Let me make sure, I only draw triangles, perhaps this one here is the first one struck and there are some other ones further away.

So, for step number two, we need to do intersection tests with triangles and then figure out the closest triangle that is struck, that is going to be the part of the scene, that this image, this pixel in the image is effectively looking at right. So, that is what we have to figure out, we need to figure out, what should the amount of intensity and color be corresponding to this particular triangle that is it. First, this other triangle that is hit later, would be occluded or completely behind drawing, where the intersection points might be. So, you can look in the book, if you like or try to derive yourself, some simple

equations or piece of code for testing the intersection between a triangle and three dimensional space and a line segment, in three dimensional space.

So, it is not too complex, some sort of, maybe high school geometry and algebra exercise. Now, of course, if you are trying, if your scene has millions of triangles in there, then, it is not very efficient to go and loop through each and every one of these triangles, for each and every one of these pixels and. So, you may want to develop a data structure that cleverly arranges the triangles. So, that you can eliminate large fractions of them from having to test them, based on what zones or what regions, these ray's appear in.

So, I do not recommend, doing the absolute simplest case, where you just loop through a million triangles in each and every one of these pixel by pixel operations right and. So, after we do that, we figure out ok. This is the part that I am going to care about. Here, let us say, this is the nearest intersection point. So, this is what is showing up right, there in the image. Again, I am just imagining the image runs this way as well, coming out of the board. So, we need to figure out at this exact location, what should we draw? What should the R G B values be for this pixel, based on the fact that this visibility ray, it is this triangle first.

Well one thing we assume is that in this virtual world that is been constructed, we want to place some light sources and then the light is going to hit our objects are triangles in this case and then based on the reflectance properties of the triangles, we are going to determine what the R G B value should be. So, that gets to the shading part, which is part three here. So, let me now go into shading.