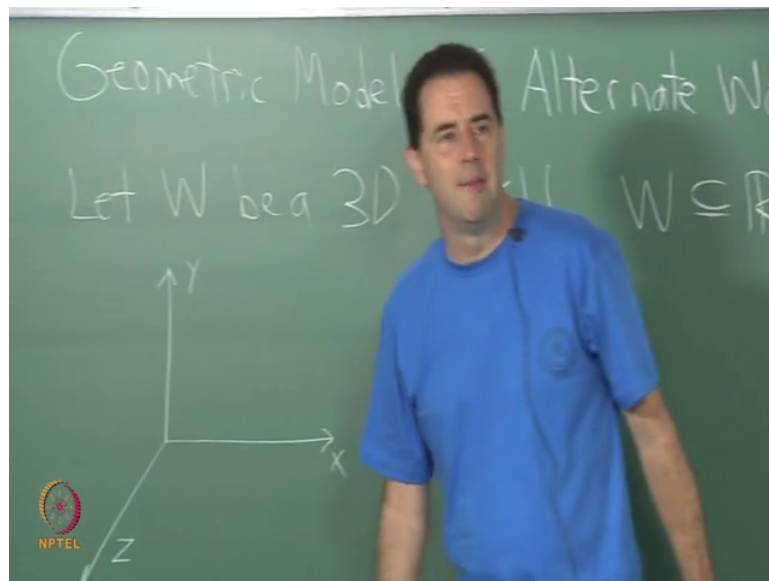


**Virtual Reality**  
**Prof. Steve Lavalle**  
**Department of Applied Mechanics**  
**Indian Institute of Technology, Madras**

**Lecture – 03**  
**Geometry of Virtual Worlds (geometric modeling)**

Hello, welcome to the next lecture. Last time I gave you an introduction to the course and I provided a Bird's eye view, which was an overview of the hardware, software and sensation and perception parts of virtual reality. So, during the software part 1 of the key things, I mentioned was the alternate world generator. So today, I want to go into the basic mathematics, the kinds of modelling and transformations that we need to do to set up an alternate world generator.

(Refer Slide Time: 00:49)

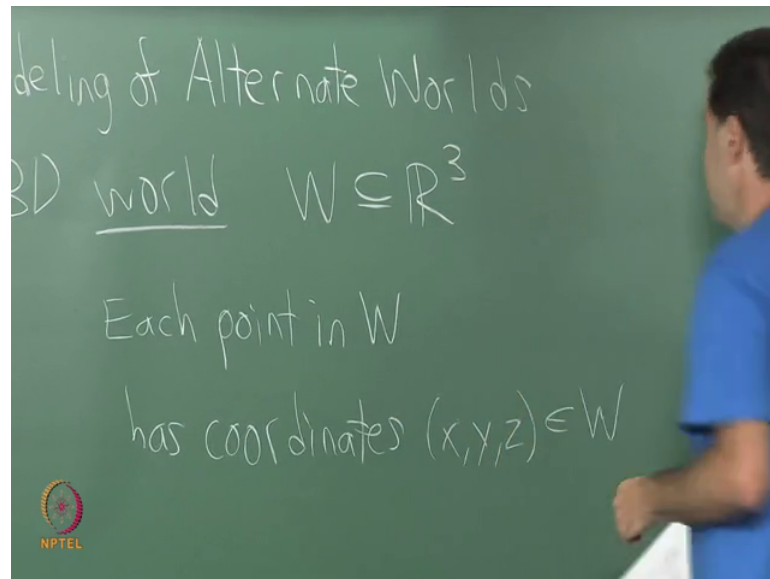


So, today's lectures will be more fundamentally oriented I will say with Geometric Modelling of alternate worlds. So, the alternate world generator, in some sense has to simulate or reconstruct some kind of world. So, it may be simulating a virtual world. It has geometry, it has physics and the geometry and physics has to look reasonable to your brain, has to appear reasonable to your brain or to your perception when you use virtual reality; when you are in this kind of experience. So, we may be simulating a virtual environment or you may be capturing a real physical environment in some way, it could

be far away or it could be recorded or both and you are transmitting that through this kind of head mounted display.

And that provides another example of virtual reality. So, we have also let  $W$  be a 3D world.

(Refer Slide Time: 02:02)



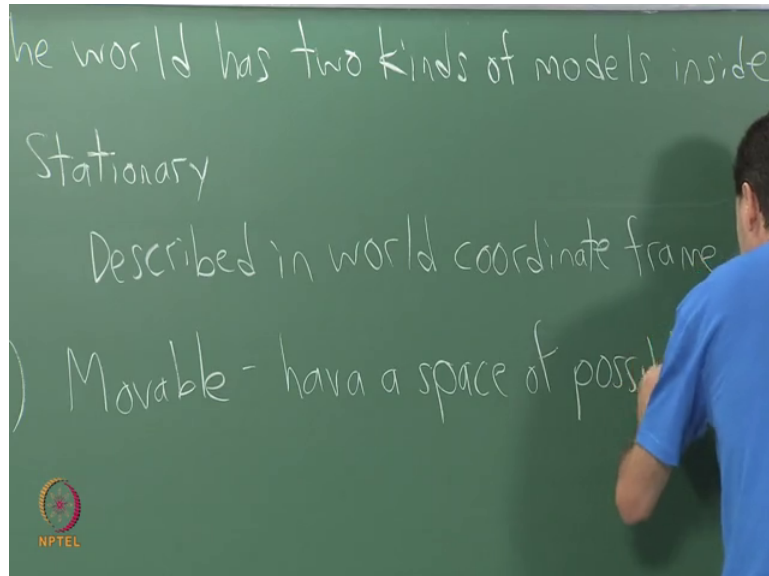
So, I mean that  $W$  is some subset of  $\mathbb{R}^3$ . So, I will just start doing mathematics in Cartesian co-ordinates on the Reals in 3 dimensions. I mean define a co-ordinate system. So, we will have  $X$  going horizontally, we will have a  $Y$  axis going upward and we have  $Z$  coming out of the board done in a way. So, that we have a right handed co-ordinate system.

So, I will prefer and use right handed coordinate systems all the way through in graphics and rendering it is sometimes preferable to use left handed coordinate systems and in direct  $X$ , if you are doing programming in that its left handed if you are doing programming in open gl a lot of other systems it is right handed, but most of math and physics is right handed oriented. So, I want to stay in math in that way; also I will follow the conventions used in the in the game industry and throughout much of graphics which is that the  $Y$  axis is up throughout most of engineering.

I think the  $Z$  axis is up, but we are going to stay consistent with that. So, that whatever you apply from game engines and related things it ends up being consistent. So, this is a

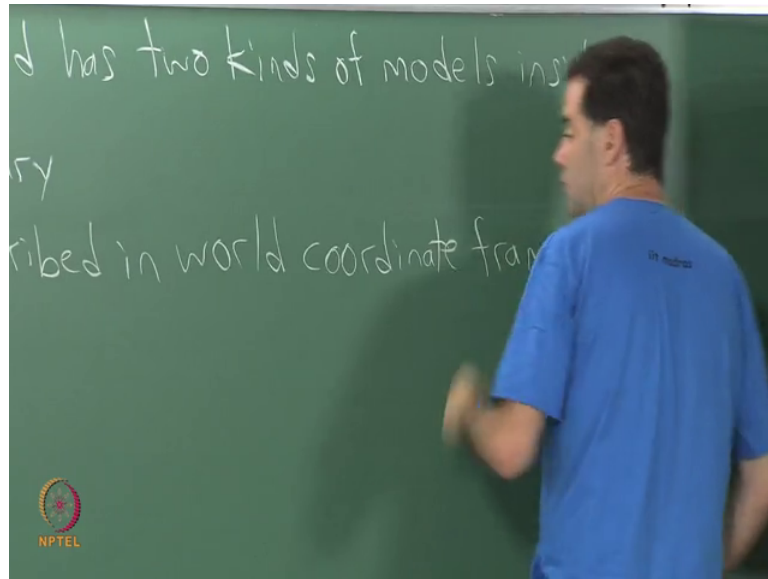
3 dimensional world and so, each point in this world has co-ordinates  $x, y, z$  element of  $W$ . So, I would just have I will be describing points with  $x, y, z$  co-ordinates.

(Refer Slide Time: 03:54)



So, the world has two kinds of models. The first main distinction I want to make here, is 1, we have Stationary models. So, this means as the name implies they never move. So, this can correspond to buildings, walls, trees assuming the trees are not swaying in the wind, so, so, so. Think about things that when you put them into the world, they remain fixed all the time. That is the simplest case; in this case they are described in the world co-ordinate frame.

(Refer Slide Time: 04:50)

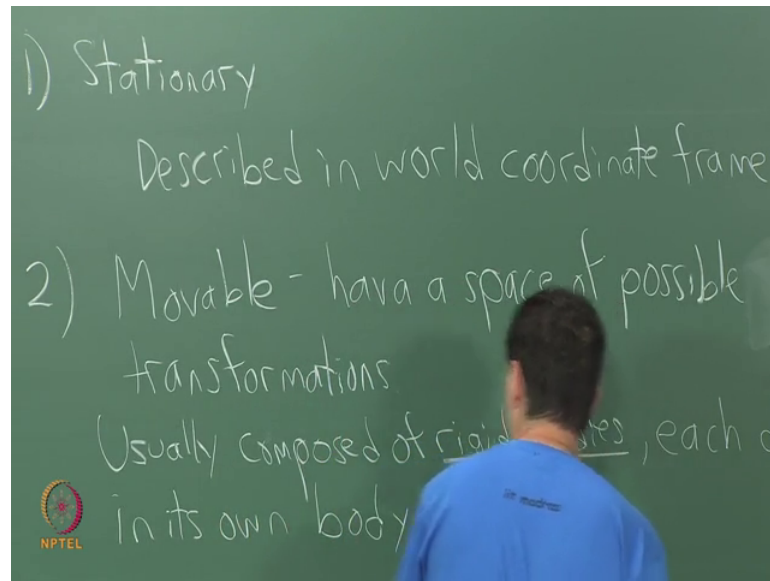


And so, we can just make up these  $x, y, z$  co-ordinates for whatever points we want to describe for the stationary models and leave them alone the whole time. One thing I should say, if you are building a virtual world, I highly recommend scaling these axes in a way that corresponds perfectly to the world you are making.

So, I highly recommend for example, if you use the Metric system; then, measure out everything in meters or centimetres whatever it is that you would like to use. So that you can always remember what the scale correspondence is between your virtual world and the real world. Unless you are doing interesting we are experimenting where you want to change your scale entirely which is interesting by itself. I would rather do it as an another transformation that you apply to this. It is very nice and helpful to keep the numbering in here, let us say consistent with measurements in the real world.

So, that if you draw a chair for example, the chair ends up being the right size, if you are trying to capture a chair from the physical world and bring it into the virtual world. You do not have to study it, look at it you end up with problems of perception of scale and it gets very difficult. So, it is nice to keep these coordinates like that, match them as they would be in the real world. So 2, the other kinds of model inside of the world are Moveable models and in this case these models have a space of possible transformations, space of possible transformations, you write that down here.

(Refer Slide Time: 06:47)



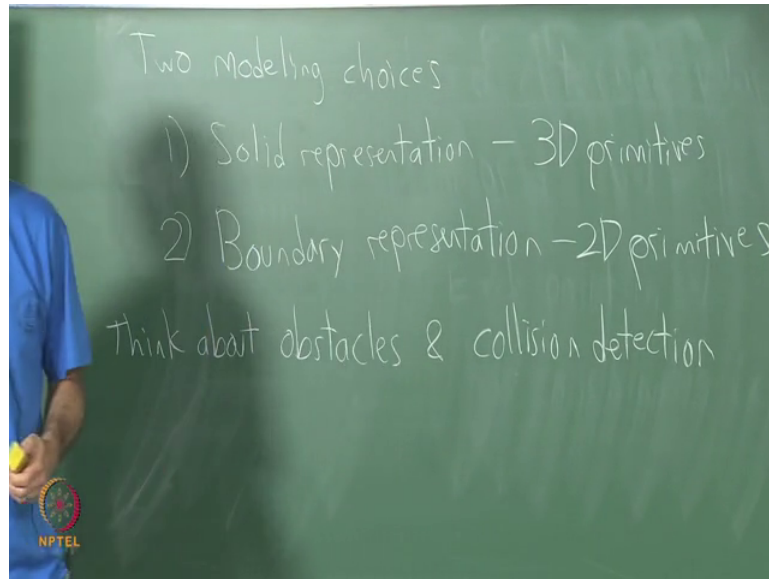
So, this may correspond to many moving things, it could be the avatars representations of yourself, anything else moving around other characters, animals, moving bullets, blowing leaves, all sorts of things and usually these are composed of rigid bodies. So, imagine having a bunch of rigid bodies attached together. So, Usually, composed of rigid bodies each of which is defined in its own body frame.

So, we will not directly use the world co-ordinate frame, but each body will have its own frame and then, we will imagine a moveable model as being a collection of 1 or more rigid bodies. So, they could be attached together just like a Human Skeleton let us say, where we have each rigid body may correspond to a particular link and the links are all joined together by joints and we end up with some kind of movable structure.

It could just be a single rigid body that simply moves around through the space maybe you have a chair that can be moved around. So, it is rigid by itself, but it is a movable body; the character may be able to come up grab the chair and move it somewhere.

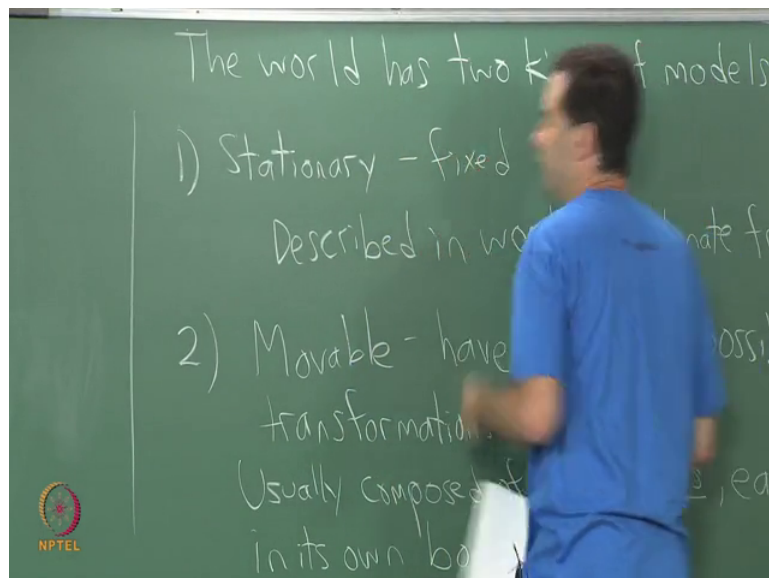
So in these cases, we have to talk about a space of possible transformations that can be applied, in order to do that we start off by describing it in the body frame.

(Refer Slide Time: 08:44)



So, we generally have two Modelling choices; when we are making geometric models of these bodies that are in the world, whether they are stationary or moveable.

(Refer Slide Time: 09:00)



Whether, they are stationary or movable. 1, choice is to have what is called a Solid representation and the 2nd one, is what's called a Boundary representation.

So, in the case of Solid representation, we have three dimensional primitives and so, the most basic description unit we have represents some 3 dimensional chunk of the world or of the body. And in the Boundary representation case, we have 2D primitives. And there

are lots of details to this, lots of interesting technical complications and a lot of trouble comes about based on which representation you choose, you solve some problems and then other problems emerge.

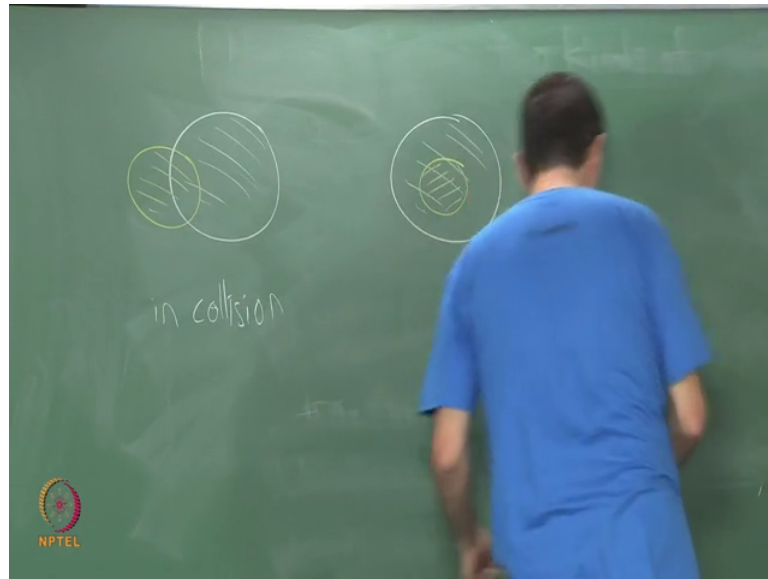
And I will give you a little bit of a hint to that of that today, but I will not be covering it in full detail. So, one thing to think about with regard to this is think about obstacles and collision detection. In the real world when I walk up to this table my motion is physically obstructed. So, that is one kind of obstacle. If I put this notebook in front of me, I also have some visual obstruction as well and you could have 2 these could be independent right.

I could have some fogg that does not obstruct my motion, but it just obstructs my visibility. I could have what is an example of an obstacle that obstructs my motion, but not visibility.

Student: Glass

Glass, very good. So, I can look through glass and I get obstructed maybe surprised by that. So, those are 2 different kinds of obstructions you could model these things and represent them. I have not talked about rendering or graphics yet; we will have to get to that part, when we talk about drawing the world in a way and presenting it drawing it in a way that is appropriate for your vision sense, we have not got to that yet, we are just describing the geometry of the world.

(Refer Slide Time: 11:34)



So, with regard to these 2 choices having 1, body let us say represented like this and I have another body that is here. In this case I would say they are in collision because they intersect right there is a non empty intersection. Now if I have a some kind of solid representation, I am only drawing this in two dimensions. Hence the solid representation looks 2D and a boundary representation should look 1D.

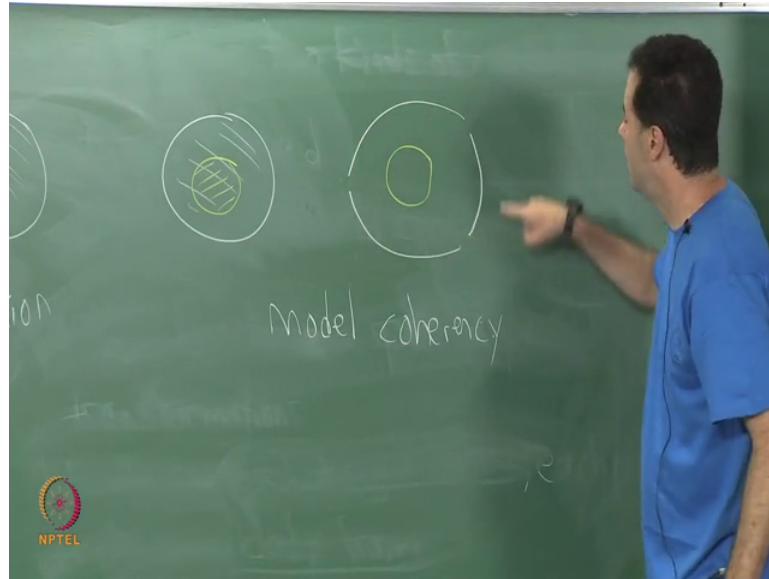
So, if I just think about representing these by their boundaries just the outer perimeter of the yellow blob and the white blob. Then these also intersect at these points here. So, if I were going to do some kind of intersection test, I could look only at the boundary maybe I break this into tiny line segments and I do line segment tests or maybe I get a little more clever with the algebra, I realize these are circles.

So, I could do some circle intersection tests, but what happens when I have this case? So, in this case if I am using a solid representation, I may be able to tell very quickly depending on how I have decided to represent it; that the yellow disc is in fact, in the interior of the white disc.

But if I have a boundary representation, there is no intersection of the boundaries. And so, it is just something very important to think about there is some kind of meaning behind these representations.



(Refer Slide Time: 13:26)



And it is somewhat implicit and then later as you start developing you realize some of the shortcomings and problems. Let me give you an example it gets a little bit worse. It may be that I have a boundary representation and it has mistakes in it a little bit.

So, this is the boundary representation that I have, of one object or body and well maybe the other one is fine and looks like that. So, that is the bound representation of that one and I am still trying to infer whether or not this yellow disc is inside or outside of the white disc.

And it may be that I have a perfectly defined disc here, but out here because I have these breaks, I am not even sure if there is a well defined notion of inside or outside. What does it mean to be inside or outside of this? How can you define that? If this is a closed curve then, by the Jordan Curve Theorem, there is an inside and an outside and that is all there is it very nicely partitions of space into 2 regions. But when there is this break in here, it is not very well defined and you can imagine that the same thing happens in 3 dimensions and it is much worse and you have to think about where does your model come from?

If you are drawing the model yourself maybe using a design tool, maybe using Maya or blender and you are making your own models perhaps you can get everything very consistent; this is called Model Coherency where this particular example I have given

has a lack of coherency, you meant this to be a closed curve, but it ends up getting broken.

So, you have to ask yourself what is the coherency or perhaps consistency of the model; if you are extracting this model using depth cameras right maybe you have computer vision algorithms running, you are also getting depth information, you are trying to build some kind of surface description. What do you think? The chances are that there is going to be little holes in your data. And that is going to make a mess.

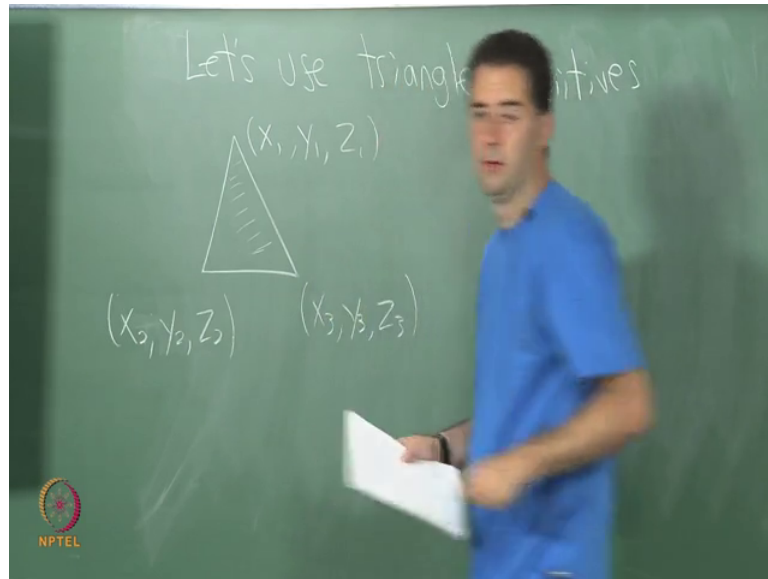
So, you may have to do a lot of work to build a coherent model you may have even more work perhaps to build a nice solid representation of this, but then when you do that then it will be easier to determine collisions. So, these are just some of the kind of difficulties you end up with also because it is difficult to know what side you are on of these boundary pieces; it gets sometimes difficult to render surfaces in computer graphics because you are not sure whether the surface normal that you are looking at is pointing outside or inside of the object to be renders a lot of difficulties with that.

If you do some rendering on your own it often mistakes come up because the surface normals are pointing in the wrong direction. So, these are some different choices at the highest level. I am going to only work with Boundary representations even though I think Solid representations are cleaner in a lot of ways. One of the reasons, why we should talk about boundary representations here is because that is what is done in primarily in game engines, all the way down to GPUs.

So, the graphical processing units that are that are in our computers are geared for boundary representations and in particular representations that are made of 3D triangles. So, why do not we pick the simplest case boundary representations and the 2 dimensional primitives are going to be triangles that sit in 3 dimensional space.

So, I call them 2D because it is a 2 dimensional triangle; it is not a 3 dimensional simplex or pyramid of some kind that is sitting in space; it is a 2 dimensional surface that sits in 3 dimensional space.

(Refer Slide Time: 16:43)



So, it is a 2D primitive. So, let us use a triangle primitive. So, in order to describe a triangle, we may mean well we generally mean this 2 dimensional surface. It is assumed to be linear as part of a plane.

And so, all you need to define are the vertices of the triangle  $x_1, y_1, z_1$ ;  $x_2, y_2, z_2$ ;  $x_3, y_3, z_3$ . So, I can start defining triangles like this. If I put them into the world, I define them in the world co-ordinate frame for stationary models and again, if they are movable models; then, we define them in the body frame and then I am going to say what to do later on how to transform those. There is also some common tricks for gluing triangles together in order to save space and to make more efficient data structures to access all of the triangles in a quick manner.

(Refer Slide Time: 18:10)



And one thing that may happen is the triangles may be glued together to form strips. So, you could have a strip of triangles and do operations on that or more generally you may find the triangles are put together in some kind of mesh and there are all sorts of interesting issues in terms of what kind of property should we maintain in that mesh in order to make the computations that we perform later.

If we are studying or let us say solving problems of rendering to determine what the lighting should be across the surface, we would like to have not too many triangles, we would also like to represent the surface very well. If it has a lot of curvature we might need to make the triangle smaller, we would also like to avoid having very very thin triangles. If we captured a surface from a depth camera, how do we build a nice representation of this and there are a lot of algorithms from the field of computational geometry.

You can study these people are very interested in starting from raw data and then building nice surface representations like this Boundary representations; especially, not having incorrect holes in the data. So, when you put these together, when you make strips or mesh if the model is coherent; then, many of the vertices are shared. So, that reduces the overall amount of data storage and there are very clearly defined adjacencies here between triangles. And so, you can make data structures with pointers that move around.

One example is a doubly connected edge list which computational geometry I did developed or you can very quickly move across and propagates computations across these surfaces that you might need to do. So, so, that is about all, I will say about basic modelling and primitives are there any questions about that.