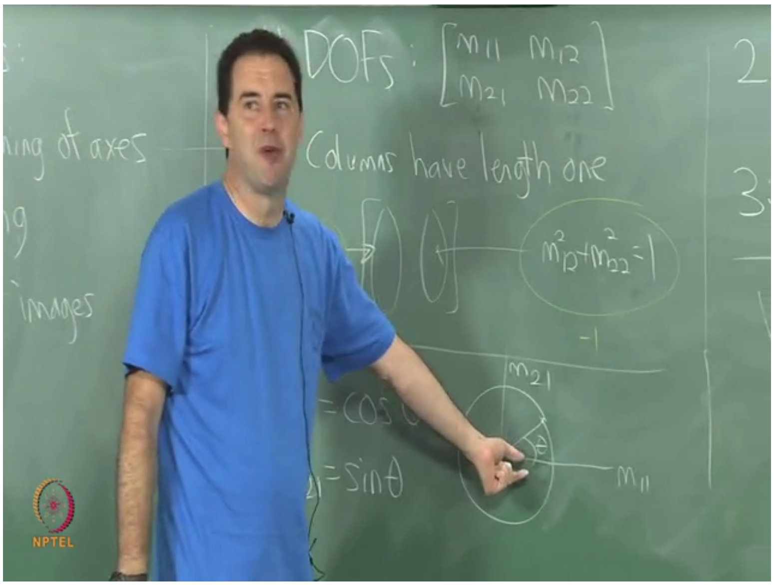
Virtual Reality
Prof. Steve Lavalle
Department of Applied Mechanics
Indian Institute of Technology, Madras

Lecture – 3-3
Geometry of Virtual Worlds (3D rotations and yaw, pitch and roll)

Questions about that. So, I am going to get to the 3D case now, which is the reason why I dragged you through all of this. One very nice property that I want to point out here because it is harder to visualize on 3 dimensions is that, I have used this parameter theta if I change theta by some small amount, say I change by a tenth of a radian or something.

(Refer Slide Time: 00:43)

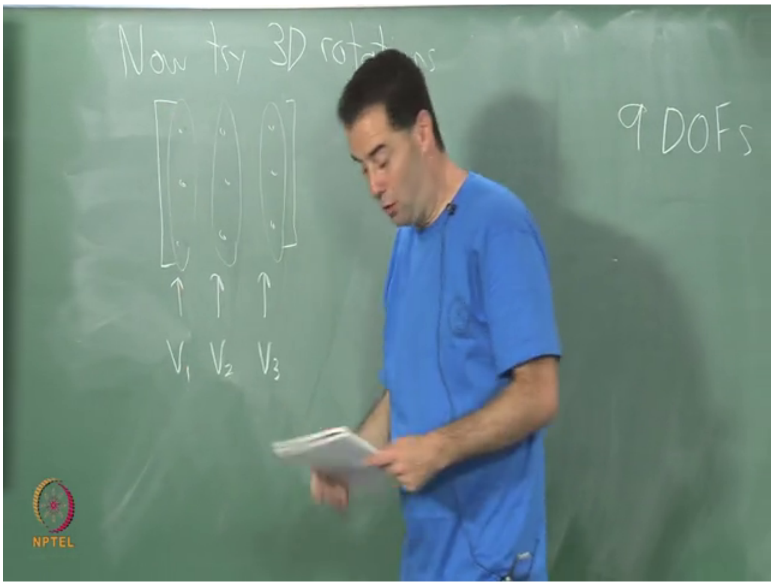


Then the amount of rotation changes by a small amount right the rigid body will move by a small amount, and it does not matter if I change by let us say if I go from 0 to 0.1 or if I go from one to 1.1 a change of point one corresponds to the same amount of change for the rigid body.

In 3 dimensions for 3D rotation it is very very hard to get that property back, until you know how to do it. But if you were just to design something yourself that has that property so, if I vary my parameters of rotation by a little bit how do I make the variation in 3D rotation be the same amount of little bit. And this is going to be very important if you do something like say design of filtering and tracking method, because you want to measure the errors in your tracking and you want to do that in a consistent way. You know otherwise you will get confused you will think you have huge errors over in this part of the rotation space and very tiny errors over in this other part when. In fact, the errors might be the same in terms of how much the body is actually rotating.

So, that is the kind of difficulties we get into in 2 D very easy to see these things it gets harder in 3D questions about this.

(Refer Slide Time: 01:43)



So, I am going to go to the 3D case. So, now, we go to try 3D rotations I do not think I feel like riding 3 by 3 matrices with a bunch of m s and subscripts and all of that. So, I will just write it with dots inside. So, we have bigger matrices is now, I will just write some dots here and make it fast you can fill in the m s if you like.

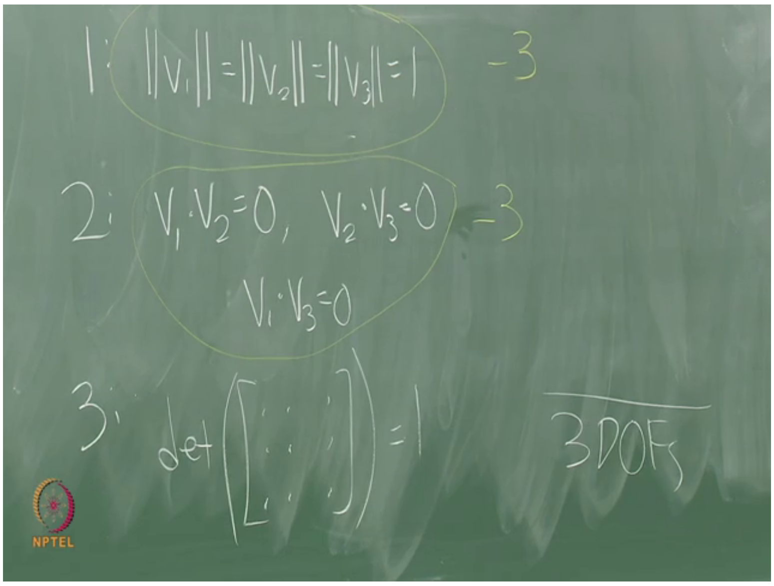
So, now I have 9 degrees of freedom for a 3 by 3 linear transformation with real valued entries here. So, we start off with 9 dots and I want to use the same kind of reasoning now, let us give names to these column vectors here. So, I will call this V 1, V 2 and v 3. So, I will use that to refer to these columns because if you remember in the 2 D case we kept referring to the columns. So, I would just refer to those as V 1, V 2, V 3 one may also want to refer them as v x, v y and v z, but I am I can be careful if I do not want to make them look like scalar subscripts.

So,. So, just take V 1, V 2, V 3 alright. So, constraint number one what was constraint number one anybody remember?

Student: (Refer Time: 03:01).

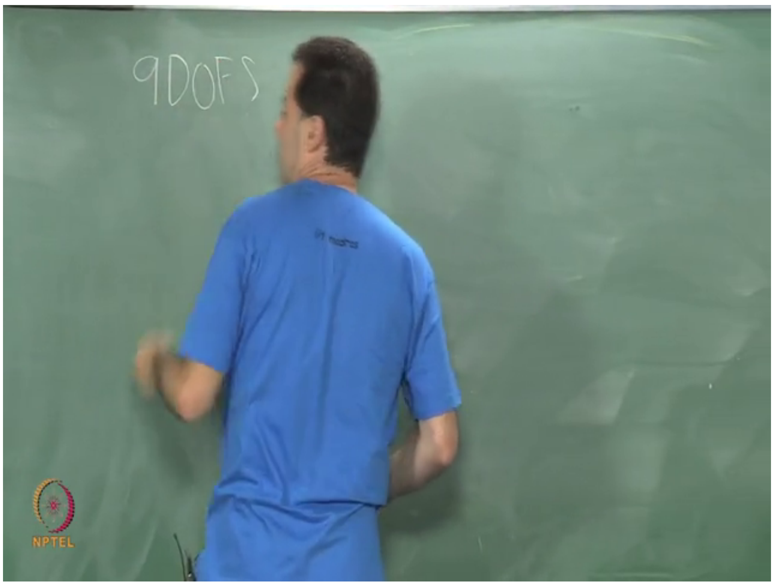
See yeah. So, the length of the vector should be 1 and. So, if I take the if I square each of the components and add them up.

(Refer Slide Time: 03:13)



I should get equal to 1. I guess I can just write that using norm notation from an algebra, but if you comfort anymore just square each element add them up. So, the norm of V 1 should be equal to the norm of V 2, should be equal to the norm of V 3, should be equal to 1.

(Refer Slide Time: 03:36)



And. So, we started off with 9 dots, how many did I just lose?

Student: (Refer Time: 03:42).

Three right. So, I lost 3 there. So, I have this reconsider this thing gives me a degree of freedom penalty of 3 you know.

So, I lost 3 degrees of freedom right 2, 2 was this thing about inner products right you have to avoid shearing. So, you have to have the inner products be equal to 0 right the dot products between the columns which columns, do I pick once that all of them all let us see well dot products there is not really a triple way product here right. So, we got to do it in pairs very good. So, we do all pairs from (Refer Time: 04:20) see if re choose 2 is equal to 3. So, there is 3 pairs it does not matter, what the ordering is because dot products do not care about ordering cross products do.

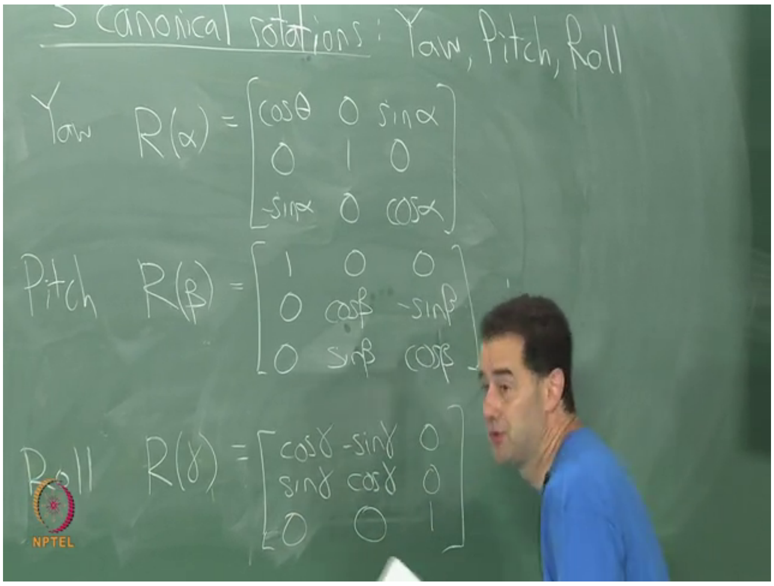
So, I guess I just need to pick the pair s here. So, I say I need to have V 1 dot V 2 equals 0, I need to have V 2 dot V 3 equals 0, and V 1, V 3 right get the other pair. So, I need to have V 1 dot V 3 equals 0. So, when I put all that together I guess I have lost 3 more degrees of freedom. Hey not bad at doing the same kind of pattern as for 2 D. So, we lost 3 more with these we still have the third condition which is on we do not want mirror images.

So, 3 we need to have the determinant of this take 3 by 3 matrix be equal to 1 and again this is not going to drop the degrees of freedom, this is going to keep it the same it is just going to eliminate half of the cases. In fact, because every mirrored transformation has a corresponding one that is unmirrored, it is getting rid of exactly half of them. So, in the 2 D case as I said there was like a good circle and a mirrored circle, there was the true rotation circle and the mirrored circle.

We now have remaining some 3 dimensional set or 3 dimensional space of rotations we have 3 degrees of freedom we have to get our fingers on it somehow describe it in some good way we have 3 degrees of freedom left and it was in 2 different components there s like the mirrored part and the unmirrored part. This constraint eliminates the unmirrored part. So, if you want to have the mirrored part then just make the determinant be negative one and. So, we are going to avoid that. So, by the time we are done we finally, have which we should have we get all the way down here past 3 constraints.

And we have 3 dots for 3 dots four 3D rotations all right.

(Refer Slide Time: 06:48)



So, I am going to start off with a simple way to describe 3D rotations, and then after that get to the more let us say appropriate way for doing the things that we want to do. So, to start off here and we will take a break shortly, let me just start off here with the 3 canonical the 3 canonical rotations which are yaw, pitch and roll. So, these are going to be 3 independent parameters we can use to describe rotations, they are the most I would say intuitive and easy for people to understand, but they cause a lot of trouble and damage later. So, I am going to first introduce them we can talk about them, they are they are they are helpful if you keep them separated, but when you combine them together you sometimes end up with a mess.

First of all make sure we all know what we are talking about. So, yaw is going to be this right let us all do this together. So, yaw is like this right good then pitch. So, yaw is like no right and pitch is like yes and then roll is like this, which if I understand head gestures around here means maybe you know. So, I am not sure is that right. So, so we have yes no and maybe. So, so let us all do this together. So, yaw say it together yaw.

Student: Yaw

Pitch.

Student: Pitch.

And roll.

Student: Roll.

Roll and. So, keep those straight it is very very helpful yaw pitch and roll. So, let me just write out the matrices for these and then we will and then we will break yaw, I will write it like this I will say it is a rotation matrix, with a parameter alpha and it looks like this cosine alpha 0 sin alpha 0 1 0 minus sin alpha 0, cosine alpha. This looks a lot like the 2 dimensional rotation matrix, if you look at the corners and the center part looks a yaw like the identity matrix right.

So, the center part looks like the identity matrix because it is basically saying do not disturb the y coordinate. If I do a yaw back and forth notice that the yeah notice that, the y coordinate is not changing right y is up I forgot my coordinates for a bit here yeah y is upward. So, notice y is not changing when I do this correct only x and z are. So, it is essentially a rotation in the x z plane, that is why it looks like the 2 D rotation matrix. This is a little bit of change with respect to it, but that is because these are appearing in the corner if you do a circular shift downward and over then it will look exactly like the 2 D rotation matrix being applied.

So, pitch is very similar we write R of beta let us say, which access is left undisturbed for pitch sorry do this x is undisturbed. So, we can already guess that the x part is going to look like an identity matrix, and then here in this remaining part this remaining block we just make the 2 by 2 rotation matrix, but with a beta cosine beta, minus sin beta sine beta cosine beta and then finally, roll; roll should leave the z coordinate alone then right. So, z is going back and forth z is the depth. So, z goes back and forth when I do a roll I should not be interfering with that, we did alpha beta gamma. So, gamma parameter represent that we get cosine gamma minus sin gamma, sin gamma, cosine gamma. So, the upper block is exactly this 2 dimensional rotation matrix.

But corresponding to the x y plane, and then the z part is the identity 0 0 1 0 0 0 1 right. So, that is what we get and now if you want to generate any other 3D rotation, you just need to pick some amount of yaw, some amount of pitch, and some amount of roll and then you can apply them sequentially and that will generate any rotation, but it might not provide and it. In fact, does not provide a beautiful representation in the sense that I mentioned where a small change in these parameters, may or may not correspond to the same amount of change in actual physical rotation, and that is the difficulties we will be getting to after the break.