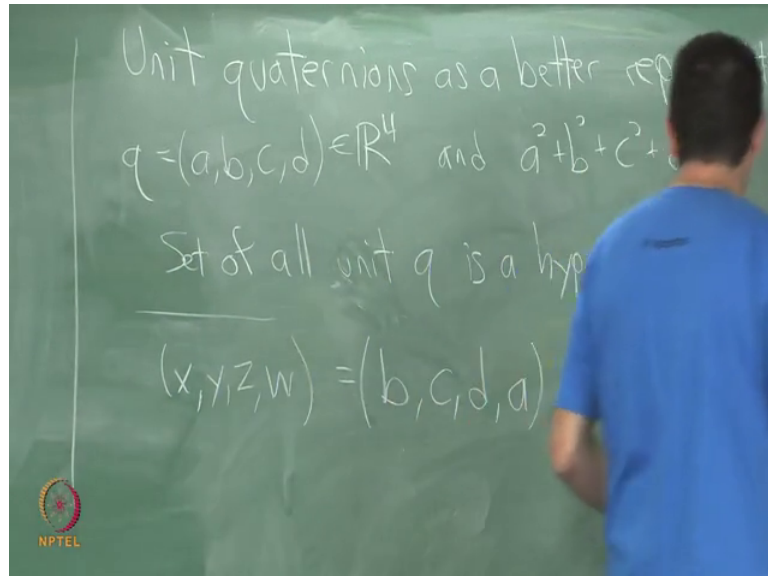


**Virtual Reality**  
**Prof. Steve Lavalle**  
**Department of General**  
**Indian Institute of Technology, Madras**

**Lecture – 4-2**  
**Geometry of Virtual Worlds (quaternions)**

(Refer Slide Time: 00:19)



So, what I am going to do is I am going to use something called unit quaternions as a better or perhaps the best representation of 3D rotation. So, I will use  $q$  for quaternion, it will look like a 4 dimensional vector. I will give it 4 very simple coordinates  $a, b, c, d$ ; not going to use  $x, y, z$  just to avoid confusion with the ordinary Euclidean space that we are in, but nevertheless you can consider this as an element of  $\mathbb{R}^4$ .

So, there is nothing strange going on with each of the components. They are just 4 real numbers 4 real parameters and I am going to impose a constraint, they are called unit quaternions which is going to mean that  $a$  squared plus  $b$  squared plus  $c$  squared plus  $d$  squared equals 1, hey that is not bad that gives us how many degrees of freedom?

Student: (Refer Time: 01:33).

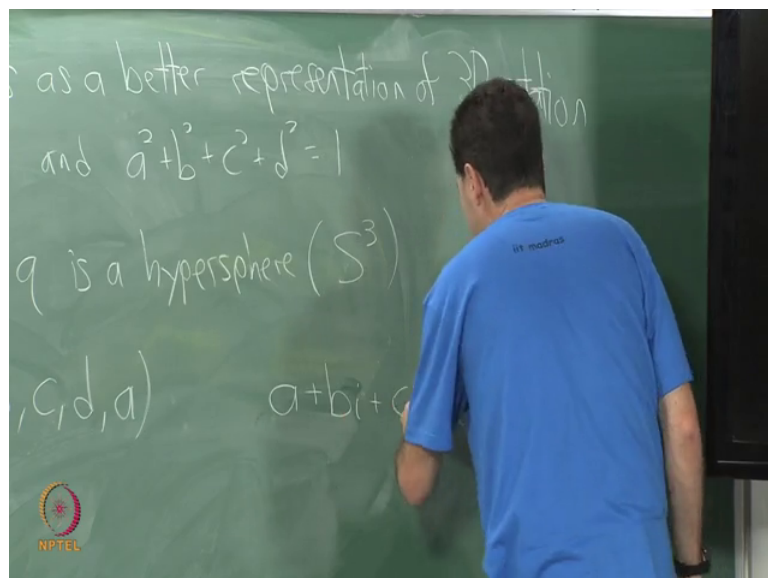
3. So, that is good. So, I just want to make sure that is a sanity check, right, just make sure, we are always keeping the degrees of freedom, right. what is this geometrically? What kind of shape is this?

It is the equation of a.

Student: (Refer Time: 01:47).

Yeah, hyper sphere or let us just say a sphere; hyper sphere that is right. So, it is one dimension higher than the 3D sphere. It is a 3 dimensional surface, right, the ordinary sphere is a 2 dimensional surface that sits in  $R^3$ . This is a 3 dimensional surface that sits in  $R^4$ , all right. So, you know; you do not have to completely visualize that. So, the set of all  $q$  that were allowed to have here the ones that are unit, let me even say it, I will say unit quaternions is a hypersphere in mathematics, it is called  $S^3$  if you are curious.

(Refer Slide Time: 02:20)



So, yes, you can do it in higher dimensions, you can have  $S^n$  in  $n$  dimensions, it is 3 because it is a 3 dimensional surface.

So, the standard sphere that you know is  $S^2$  and a circle is  $S^1$ . I should point out that you know, I am going to use this  $a, b, c, D$  representation and in unity 3 D and in dimensions, they tend to use a different representation which is  $x, y, z$  and  $w$ . So, even without being sorted alphabetically correctly and in terms of my representation which I am going to use the one that is consistent with mathematics and most of engineering as well which is  $b c D a$ . So, if you want to go back and forth between what I am doing in class and what you might see in unity and other game engines, then you have to take the first element and put it at the end.

I do not want to tell how many bugs I had in my code because of this extra change. So, it is very easy to make mistakes with this. So, so to be very careful; very commonly you will see quaternions especially in math written like this instead of writing it as a 4D vector you will have  $a + b\hat{i} + c\hat{j} + d\hat{k}$  perhaps with little hats on these, but not necessarily; so, maybe with or without the hats.

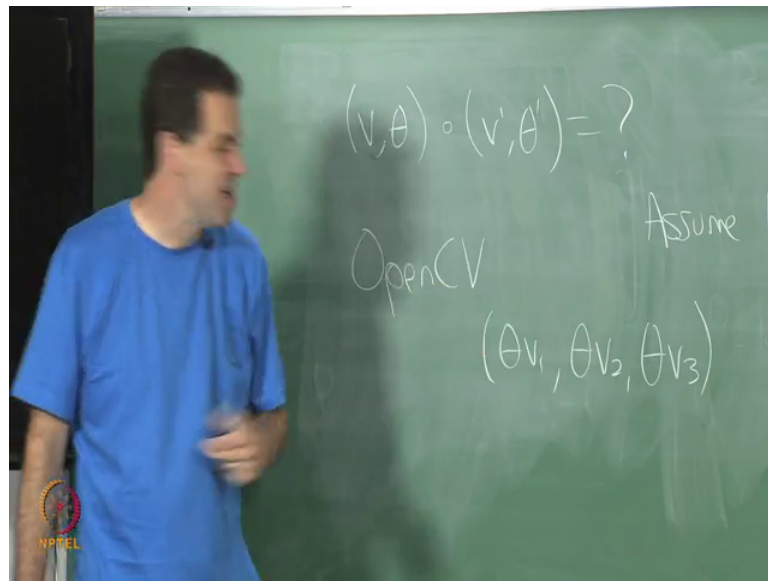
In fact, I am going to take the hats off, let me; I am sorry. So, very often you will see it written like that, if I look at it with just this part  $a + b\hat{i}$ ; that reminds you of what? Complex numbers, right maybe you put a  $\hat{j}$  there, if you are an electrical engineer because somebody used  $i$  for currents, right. So, very often people say that there is a real part of the quaternion which is the  $a$  and then 3 imaginary parts which are which are  $\hat{i}$ ,  $\hat{j}$  and  $\hat{k}$  and then this leads to lots of fear and confusion among students right and so when you deal with complex numbers and you have the real and imaginary parts, those were invented because people wanted to find roots of polynomials, right.

The polynomials have real coefficients and if you want to find all their roots, you need to invent complex numbers to have a kind of algebraic closure. This does not correspond to the algebraic closure of some kind of polynomials. So, there are some reasons for referring to these as imaginary parts because algebraically, they behave similar to complex numbers and become a kind of generalization, but they do not correspond to roots of polynomials and we are not going to do things like that.

So, in this class, I prefer not to make a big deal out of real and imaginary. I do not think that is important here. It is important for you to just think of it as 4 parameters. It is a 4D vector and we are going to normalize them. So, that they are on a sphere. So, I would say, avoid this kind of you know; let us say fear of something perhaps, nightmarish and algebraic that is happening here, if you find these things interesting go and read about it and learn about it, it is quite fine, I find it interesting, but we do not need to go down that path and so, it is important here just to keep in mind that there are 4 parameters  $a$ ,  $b$ ,  $c$ ,  $d$  and if you treat it as a 4D vector the length is 1.

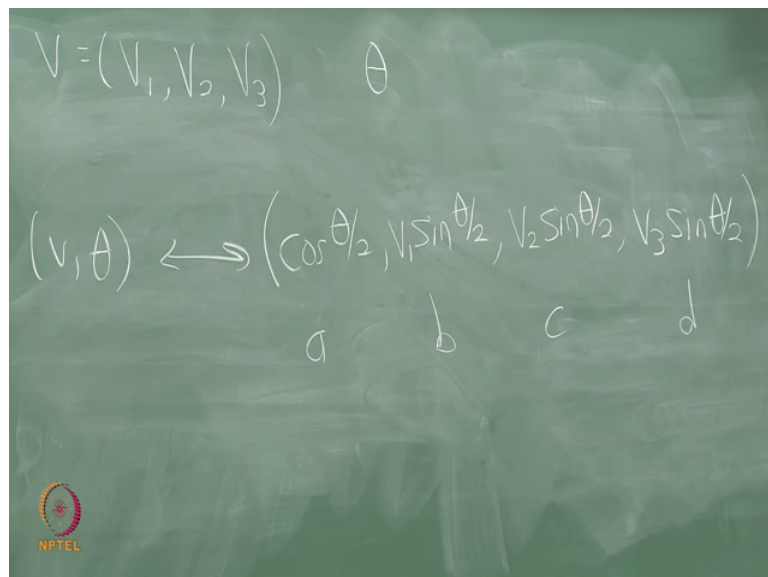
The next thing I want to tell you is how to encode a 3D rotation using our  $a$ ,  $b$ ,  $c$ ,  $d$  parameters.

(Refer Slide Time: 06:05)



So, here I gave you an example of how to encode rotation using exponential coordinates, I just took the theta and multiplied it by the v. I am going to do some other transformation or operation to do the encoding as a quaternion. So, I am going to have V and theta and convert it to a quaternion.

(Refer Slide Time: 06:30)



So, here is the thing, again suppose, I get V equals V 1, V 2, V 3 and I have theta, right.

So, I have 4 parameters that I am using. So, I want to go between V and theta and I am going to have this representation cosine theta over 2 that is the a part I am just going to

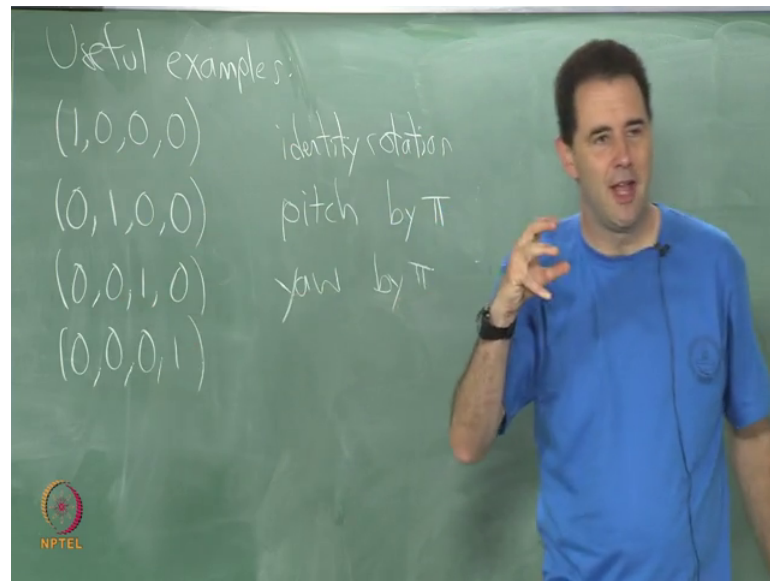
be a 4 D vector here that is the a part, the b part is going to be  $V \cos \theta$  and the c part is going to be  $V \sin \theta$ . So, that is the b  $V \cos \theta$  and the c  $V \sin \theta$  that is the D, all right.

So, all I did was write  $\cos \theta$  here and then for the remaining 3 components I just took the V vector and did a scalar multiplication times  $\sin \theta$  correct that is all I did if I square all of these and add them up what do I get right. So, I get  $\cos^2 \theta$  and then these 3 well I get a bunch of  $\sin^2$  to add up, but I know the length is going to be one because the Vs when I square those and add them up they all add up to one. So, the entire thing is going to when I square all of these is going to add up to be  $\sin^2 \theta$  by the simple trig identity I get one.

So, this is a parameterization that puts points on this unit sphere in 4 dimensional space, right. So, what I want you to get comfortable with is just being able to move back and forth between these 2. So, that when you see this a, b, c, d vector somewhere in the code maybe you are looking at unity or you are reading the code except remember that it x, y, z, w there, but whenever you whenever you see it you should be able to look at it directly and understand how to extract the angle  $\theta$  and the axis V.

So, it should be very natural for you do not have fear about imaginary parts and other kinds of strange things going on and visualization of 4 dimensional spheres and the spheres in  $R^4$  and things like that. So, 3 dimensional sphere and  $R^4$  do not worry about that just worry about understanding that there is an axis at an angle that is encoded in your and it is provided by this simple formula.

(Refer Slide Time: 09:23)



So, if you understand that let us try to go over some simple examples.

So, these are some useful examples. So, what does that quaternion do if I apply this formula; what does it correspond to?

Student: (Refer Time: 09:46).

What is that; yes. So, this is this should do this should be the identity rotation right it should not do anything if theta equals 0 then all of these components become 0 b c and D and this component becomes one correct. So, this one means the identity rotation it is very interesting that in that case I cannot even tell what the axis was anymore right because they all became 0, does that matter? No, it is the perfect case, it did not rotate anyway. So, there is no axis. So, it seems like the axis just vanishes exactly in the place where there was no rotation occurring.

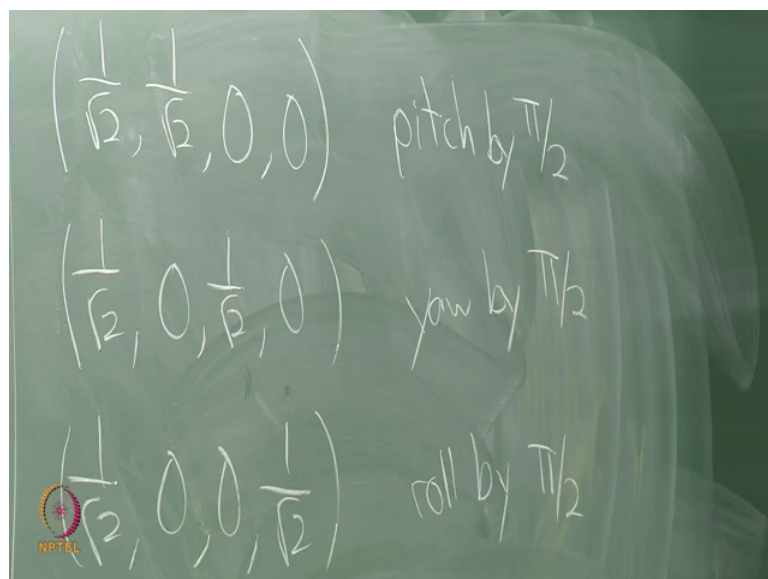
So, very nice, feels good, does not it; let us see, well; let us try some other very obvious patterns here what does that correspond to. So, that is right, 180 degree rotation about the x axis, right because the axis here for this case must be V equals 1 0 0, right, for this case. So, this is pitch, am I still being consistent without my definitions pitch is rotation about x correct. So, pitch by pi 180 degrees, well, if you know that case should be able to fill in these, this one is yaw by pi, right because it is rotation about the y axis, see how

you can look at the last 3 components and if it is simple enough you know the axis of rotation.

In fact, this is just the axis of rotation, but not correctly normalized, right, if you just take the last 3 components and renormalize them, you will get the axis you can you normalize it in this case, but that is because there is no rotation anyway. So, the axis does not matter, but the other cases, if you test for being non-zero, then you can figure out the axis very easily with your eyes not even writing code all right the last one is roll by pi right it needs a very simple example.

So, there really should not be fear of quaternions, I see a lot of fear of quaternion is this it is quite simple once you understand what to look for the first coordinate is the amount of rotation which is inverted right to smaller the number the bigger of the rotation and then the last 3 coordinates are giving information about the axis of rotation let us try another 1; 1 over square root 2, 1 over square root 2 0 0.

(Refer Slide Time: 12:43)



So, suppose I see that what is the axis of rotation?

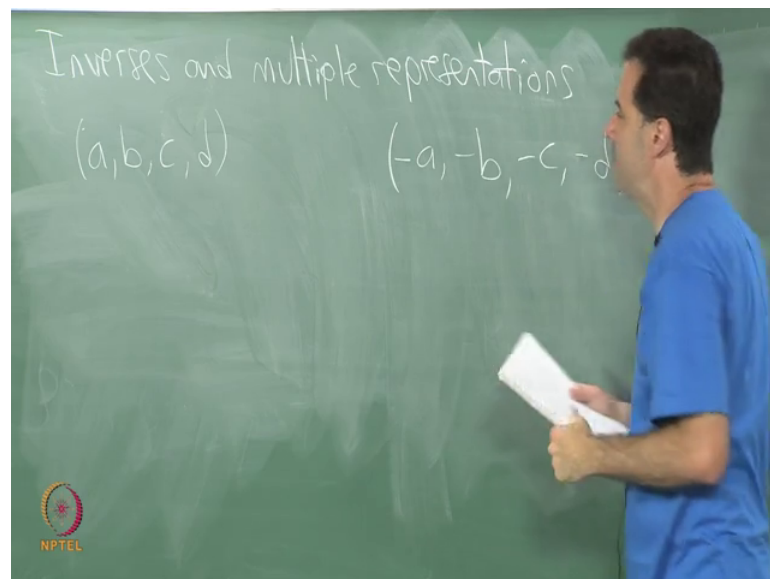
So, again that that is a perfect case where now I have to renormalize it, but it looks like it is about the x axis how far am I rotated yeah very good. So, that is good. So, you are doing you are doing great here. So, 45 or 90.

Student: 90.

90 because in the formula here, we divide by 2 so that is how some of you got fooled, right, I make the mistake a lot myself. So, it is no problem. So, pitch by  $\pi$  over 2. So, I should be able to very quickly go with the other cases one over square root 2 0 1 over square root 2 0 0 1 over square root 2. So, this is very nice if you are debugging your code, you just want to make some simple transformations and try them out it is very nice to be able to just immediately specify some simple transformations like this or if you are debugging and you are not sure you got the code correct. So, it is just very nice to be able to read these quickly.

So, this one is yaw by  $\pi$  over 2 and roll by  $\pi$  over 2 questions about that I am going to give a little more information as we as we go along here, but I want to make sure that is clear right. So, now, questions see their means it is very clear or very confusing.

(Refer Slide Time: 14:34)

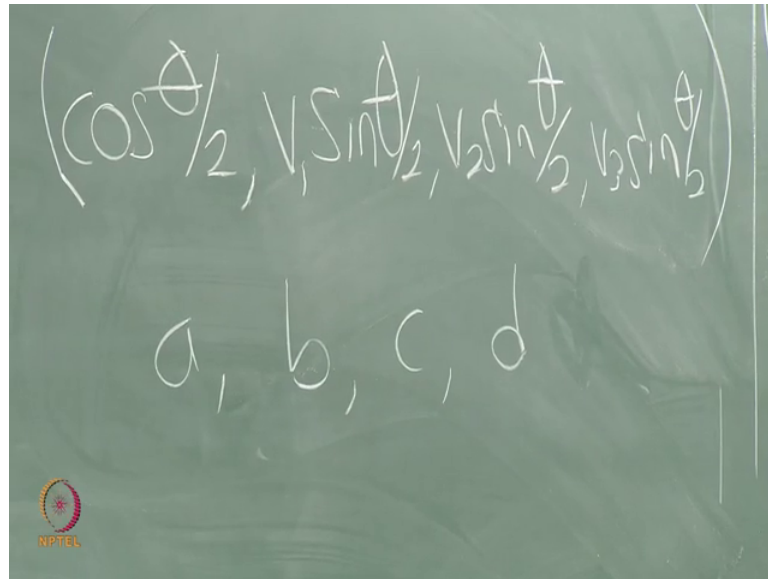


So, here is something else to think about inverses and multiple representations in other words non uniqueness multiple representations.

So, if I have a quaternion  $a, b, c, d$ . Now unit quaternion that is being used to represent a rotation I should probably write back the conversion formula.



(Refer Slide Time: 15:28)



Let me remind you of this cosine theta over 2  $V_1$  sin theta over 2  $V_2$  sin theta over 2  $V_3$  sin theta over 2, all right. So, remember that that is how we got the a, b, c, d components. So, what if I have minus a minus b minus c minus d can anyone tell me how the 2 of these are related.

So, if a, b, c, d corresponds to some 3 D rotation; what is minus a minus b minus c minus d going to correspond to is that the inverse rotation, it is the same rotation is it not that amazing, it is the same rotation because if I take the  $V$  and I twist it in the opposite direction, right, I just flip it around I will negate the b, c, d components and then if I negate the  $D$  theta as well that will negate the a, but it will not force another negation here.

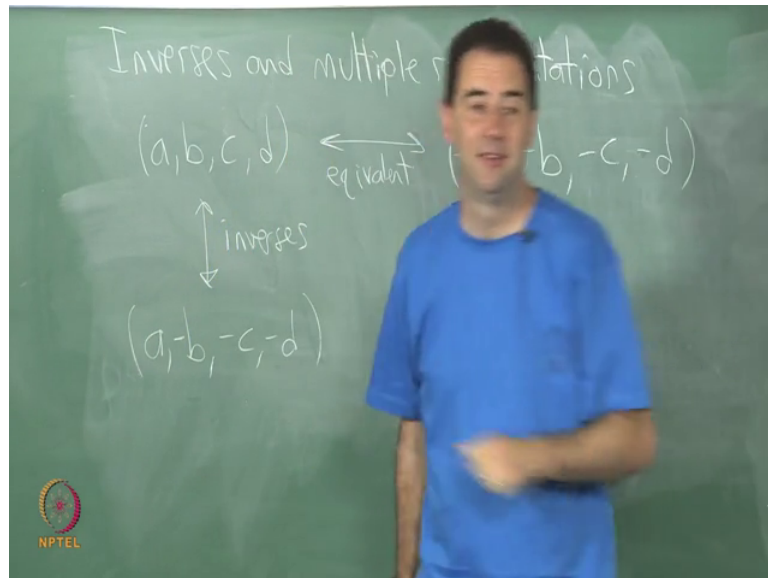
So, that will negate all 4 components so; that means that these 2 are equivalent. So, that is something very interesting to pay attention to so; that means, that when we travel on this sphere right it is a 3 dimensional sphere in 4 dimensions if we go to the opposite side of it, right, it is called the antipodal point right the opposite point what anybody know where the opposite point on the earth is from here.

Student: Yes.

Maybe it is a good guess I do not know no it has to be in the southern hemisphere. So, you have to change hemispheres in all cases right. So, you know you go to the opposite

point on the earth and that corresponds to the antipodal point. So, on this quaternion sphere this unit sphere of quaternions unit hyper sphere of quaternions the opposite point is always equivalent is not that interesting well just to add an extra layer of confusion, all right.

(Refer Slide Time: 17:43)



Let us think about what this one does; what if I have a, but I decide to negate the b, c and d parts.

So, I keep a the same binding gate these 3, what does that correspond to?

Student: I just (Refer Time: 17:55).

Yeah, I decided not to know;

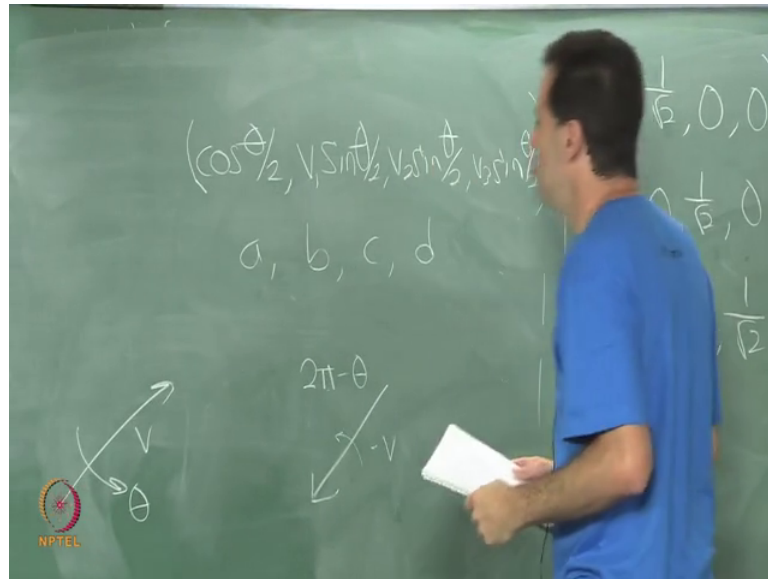
Student: (Refer Time: 17:58).

Yeah;

Student: (Refer Time: 17:59).

If I take, what do I get if I do cosine minus theta. It negates, right or it stays the same well he should say the same ok I see. So, if I move the axis backwards; let us see. So, these should stay the same, right.

(Refer Slide Time: 18:31)



So, let me think about this, if let us go back here, say I have  $V$ , I do rotation by  $\theta$ , I want to now do  $-V$  and the question is let us see; I want to make this correspond to the exact same rotation see from going in the opposite direction yeah that.

Student: I think (Refer Time: 18:59).

Yeah, it is a difficult problem of orientation. You have to visualize it. We have to remain going counterclockwise, right. So, if I want to continue going counterclockwise and apply the exact same rotation; is it going to be  $\theta$  I minus  $\theta$ ?

Student: (Refer Time: 19:17).

What is that?

Student: (Refer Time: 19:19).

$2\pi$  minus  $\theta$ , thank you. So, it is  $2\pi$  minus  $\theta$ . So, we go in the opposite direction it is  $2\pi$  minus  $\theta$ , all right. So, if I go  $2\pi$  minus  $\theta$ , what sign do I get over here? Negative  $a$ , all right, good. So, let us keep that we keep the trig going, all right. So, we are fine. Now what if I decided to do it this way instead and I did not negate the  $a$  how are these 2 related. So, if I say by Euler's 3 D representation theorem that every rotation corresponds to a single rotation about some axis, then if I want to undo that rotation I just

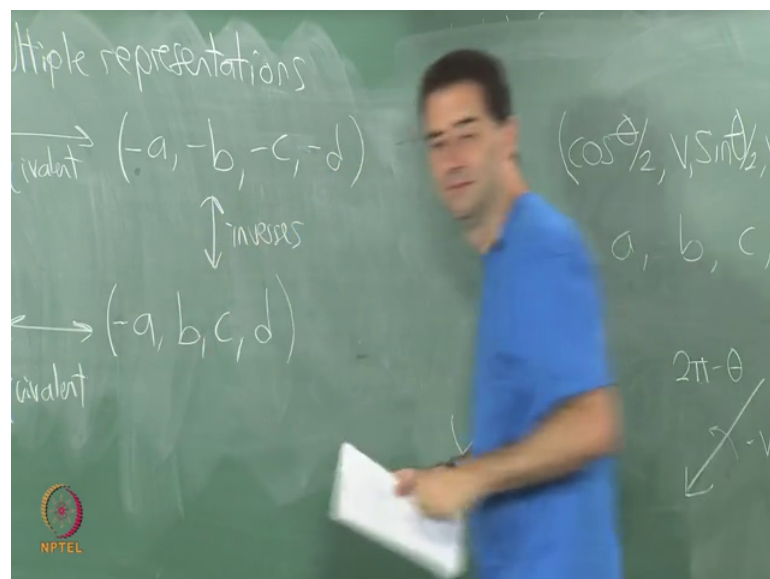
twist it back the other way right. So, that corresponds to the inverse does that make sense.

So, if I have some axis  $V$  and I rotate it by somehow  $\theta$  if I now rotate by  $-\theta$  it will just undo that rotation. So, in this case this should change the sign and I leave oh yeah this should change this sign. So, I end up getting the opposite right. So, if I keep the axis the same, but I just negate the  $\theta$  then I will negate these 3 components, but the  $a$  will end up being the same.

So, here if I relate these to its inverses is that fine. So, if you are looking at a quaternion  $a, b, c, d$  and you want to figure out what its inverse is what do you do just negate the last 3 components is that interesting it is that easy to compute the inverse rotation if you want to do it with a matrix, what do you do? Do you have to compute a full matrix inverse maybe except if it is a rotation matrix? It is just a transpose turns out. So, it is easier it is also reasonably easy in that case.

If this is the case here, then we also can have another case where I just negate the first coordinate I negate  $a$  and I leave  $b, c$  and  $d$  alone and these 2 should be equivalent, right, does that make sense? The same rule is up here if I negate all 4 components, I should end up with these being equivalent and these 2 being inverses that seem, all right.

(Refer Slide Time: 21:42)

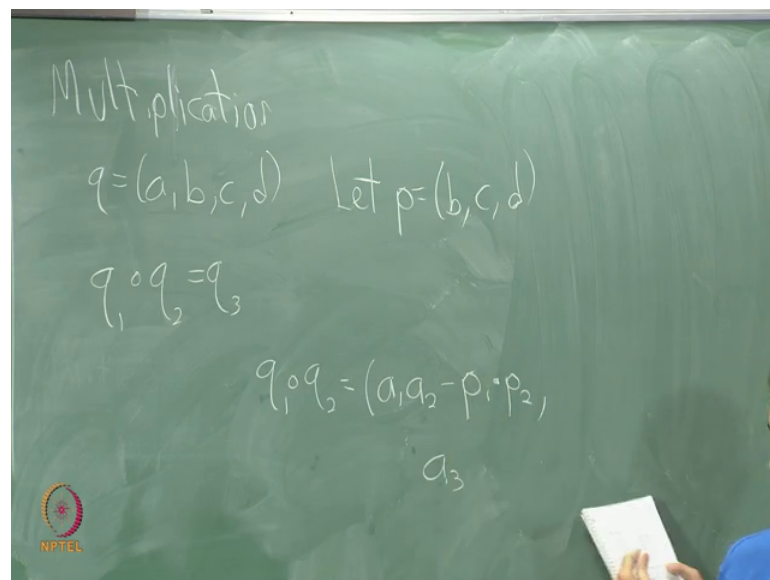


So, again just a simplest to summarize if I negate all 4 components it is the same rotation if I decide to negate only the first component or I decide to negate the second third and fourth components together, but not the first one either one of those corresponds to an inverse.

Then nice, so that is one very quick way many times in programming in these kinds of systems. It is easy to make a mistake where you should have the inverse rotation and you instead calculated the forward rotation; in that case you just perform some simple sign changes you should of course, go back and check all of your math carefully and make sure you understand what you are trying to do, but for debugging purposes, it is very helpful to know, you can very quickly spot that you are off by inverse if the signs are wrong. So, that is why I am telling you this trying to give you this extra information all right any questions about that.

There is a direct formula for multiplication of quaternions that do not that does not require you to convert back to rotation matrices I am not going to make a big deal out of it.

(Refer Slide Time: 23:07)

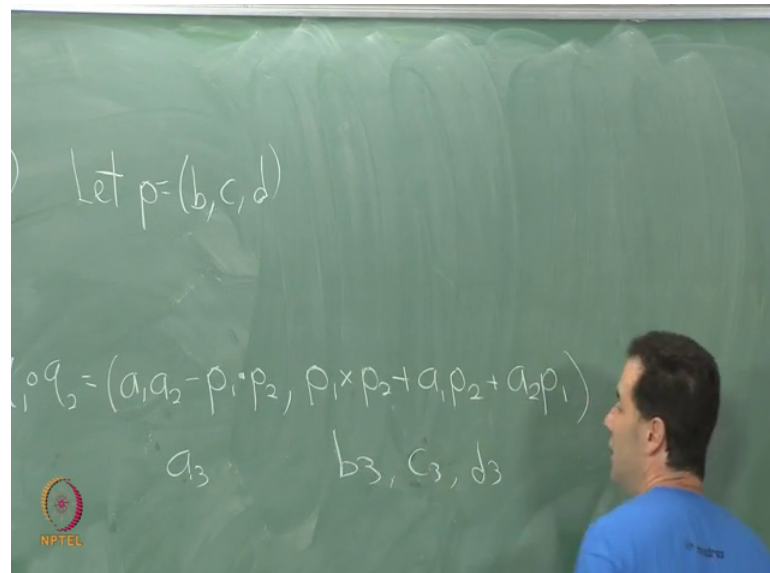


But I at least want to quickly give it to you. So, you know multiplication; right multiplication.

So, let us suppose that again, we have  $a, b, c, d$  for quaternions and I will say let  $p$  equal just  $a, b, c, d$  part. So, I can treat it as a 3 D vector. Let us just suppose that is a 3 D vector it called  $p$  and now if I want to calculate some quaternion  $q_1$  multiplied by some quaternion  $q_2$  to get some third quaternion  $q_3$ , there is a very simple formula just using 3 D standard vector calculus I say  $q_1$  multiplied by  $q_2$  is equal to  $a_1, a_2$  that is the scalar part.

The first components of each 1 minus  $p_1$  dot product  $p_2$  so that gives me the first component, now this is actually a 3 that we are calculating.

(Refer Slide Time: 24:33)

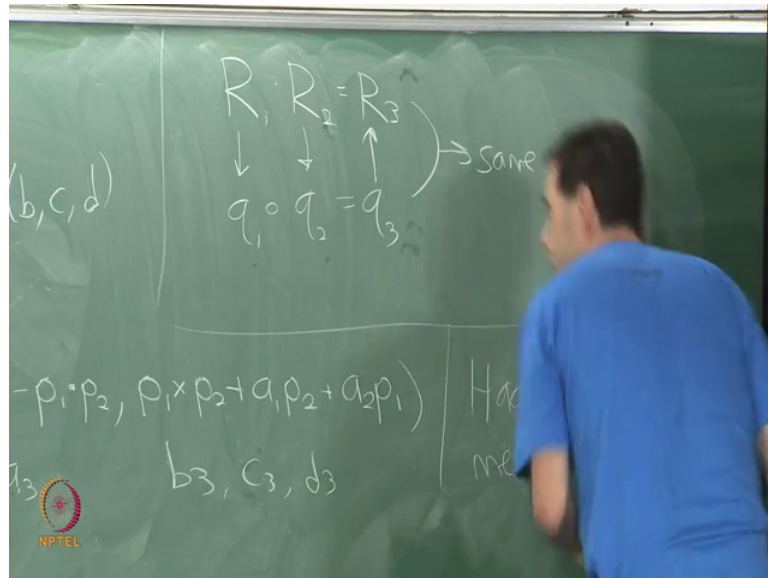


And then the remaining 3 components are given by  $p_1$  standard cross product  $p_2$  plus scalar  $a_1$  multiplied by  $p_2$  plus scalar  $a_2$  multiplied by vector  $p_1$ . So, this gives me the  $b_3, c_3$  and  $d_3$  components.

I am not deriving this at all there is a whole algebra of quaternions, there is a reason for it coming out this way, but what I want to say that is very interesting about a couple of things one of them is that if you go through this you will find out that it is not commutative this product is that a good thing it is a good thing to guess if you are using this to represent 3 D rotations and 3 D rotations are not commutative it would make sense that this algebra of multiplying quaternions also ends up being non commutative as does matrix multiplication of rotations encoded that way the reason why I will put this and you usually, you will find this inside of a quaternion library you know that you do

not have to really worry about the details of it, but one of the reasons why I put this is. So, that you know that you can do the following.

(Refer Slide Time: 25:43)



You could have rotation matrices  $R_1$  and  $R_2$  and you might want to multiply them to get some third matrix  $R_3$ .

The thing you could do is you could represent the first rotation as a quaternion  $q_1$  you can represent the second rotation as a quaternion  $q_2$  and if you do this and you apply this product that I have given here and you get a result  $q_3$  the interesting thing is that these 2 results here end up being the same if I go and convert back. So, the arrows here correspond to the standard conversion formula to convert rotations. So, if I go and I convert from rotation matrix to quaternion another rotation matrix to quaternion I do the quaternion multiplication and then I convert that back to a rotation matrix, I will get the same result and you will actually get it with fewer algebraic operations.

So, people tend to prefer to just stay in quaternion land the whole time let us say right. So, you do not ever have to convert back to matrices if you do not want to if you are using a library that forces you to in your code then. So, be it, but I just wanted to point out that because of this nice algebra you can stay entirely inside of quaternion representation. So, it is another reason why people like using them in computer graphics and in virtual reality.

In addition to this property that I mentioned, but I did not prove formally in any way which is that small changes in your in your quaternion parameters correspond to the same small changes in rotation of the rigid body regardless of where that rotations occurring at in the space of rotations and that that is very important another way to say it is that if I were to pick a an orientation uniformly at random what would that mean if it is a 2 D orientation you just pick a number between 0 and  $2\pi$  had random and you would be fine what would I pick at random for 3 D rotation.

If you pick Euler's angles the all pitch and roll at random you will not end up with a uniform covering of orientations if you pick a unit quaternion at random a random point on the sphere you will. In fact, have perfect uniform coverage. So, so if you do probabilities over transformations as well, it is very important to have this if you are interested in a lot further detail on this subject it is called Haar measure, if you want to study the mathematics of these things.