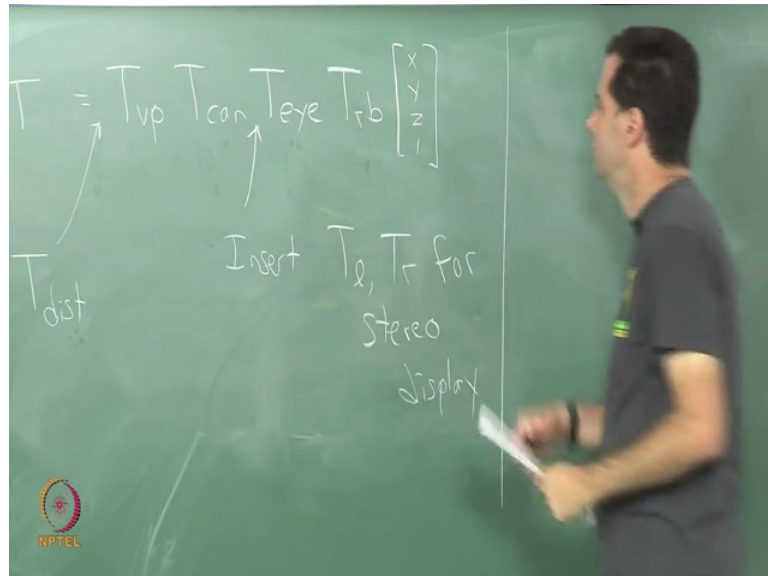


Virtual Reality
Prof. Steve Lavalle
Department of Applied Mathematics
Indian Institute of Technology, madras

Lecture – 5-3
Geometry of Virtual Worlds (eye transforms)

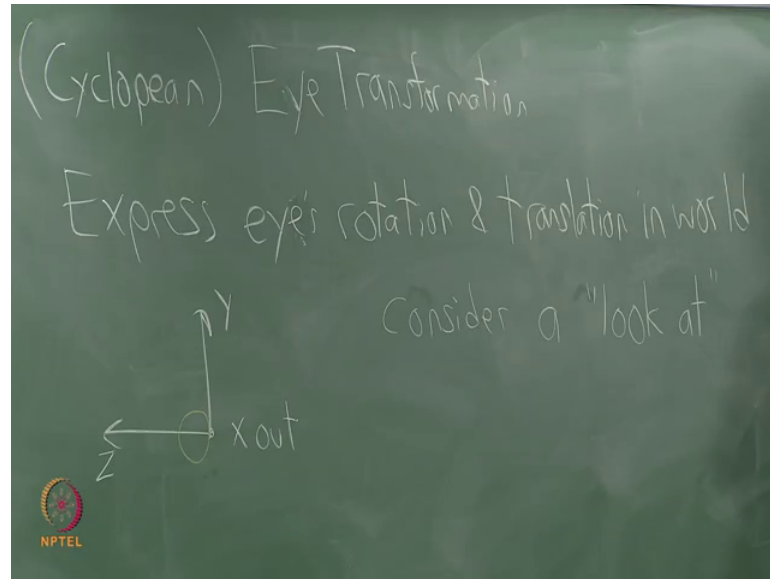
(Refer Slide Time: 00:15)



So, I would like to go into this eye transformation next if there are no questions about this, remember all of this is being applied to these points. So, if I have a you know X, Y, Z , one here right X, Y, Z is the point that I want to apply that is like a point on this cube that I drew the yellow cube that was drawn in the in the previous figure that I made on the board.

I just put the point here add a one to it apply all of these transformations and then I will figure out in some kind of pixel coordinates that look like floating point numbers or real numbers depending on whether you are using a computer or not it will it will give you these coordinates that are not necessarily integers, alright. So, I want to figure out the eye transformation.

(Refer Slide Time: 01:02)



And make it really clear I will call it cyclopean; cyclopean eye transformation.

So, if you are trying to imagine maybe people do not make a distinction in computer graphics about this, but if you are doing virtual reality, then you could build a virtual reality headset for a Cyclops and then this would be just fine right, but because you have 2 eyes, I do not want to pick the right or the left, I would rather pick the midpoint between them to make everything simple here and then talk about how to do a nice offset for left and right. So, that is why we will imagine that we are designing a virtual reality headset for a Cyclops in the in the beginning here, all right.

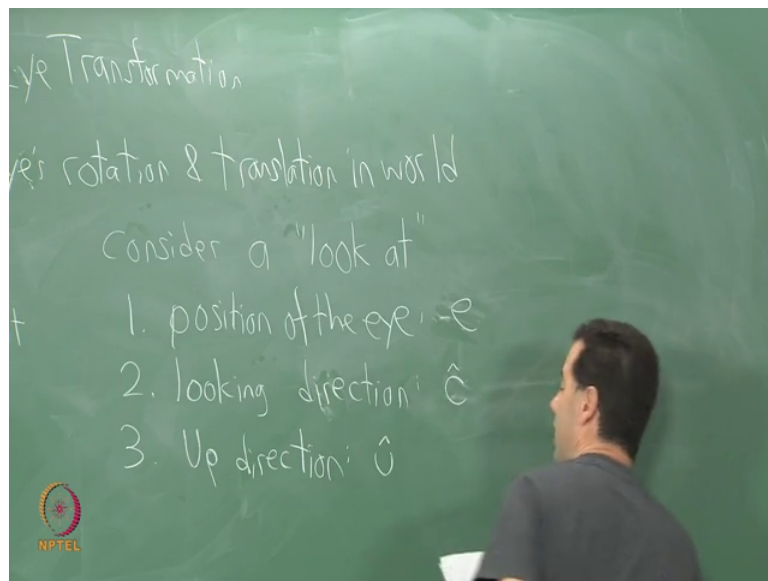
So, the first thing to do is to imagine that lets say to express the eyes rotation and translation in the world. So, in other words, I just imagined that this eye looks like an eyeball with a with a pupil is defined in some kind of body frame let me give you what is essentially the body frame of the eye, I want to put the pupil right at the origin because that is the point where the light comes in the real world. So, that is this is all virtual that I am drawing here, but never the less why I have z running into the eye and the only reason why were doing that is. So, that we can maintain right handed coordinate system. So, it is another one of these annoyances that it is here for a good reason because if you do not do it this way then you will have more annoyances.

So, it is a just its a trade off and remember x is out. So, this is essentially a body frame for the eyeball and now if I take the eyeball and place it into the world there is going to

be a transformation for that that is going to be a rotation and a translation of the eyeball, I want to figure out; what that transformation is based on how I place the eyeball and then I want to use that to make the eye transform.

So, one way to do that which is very common is to consider what is called a look at look at somewhat strange name, but a look at. So, here is how this works we have three things.

(Refer Slide Time: 03:58)

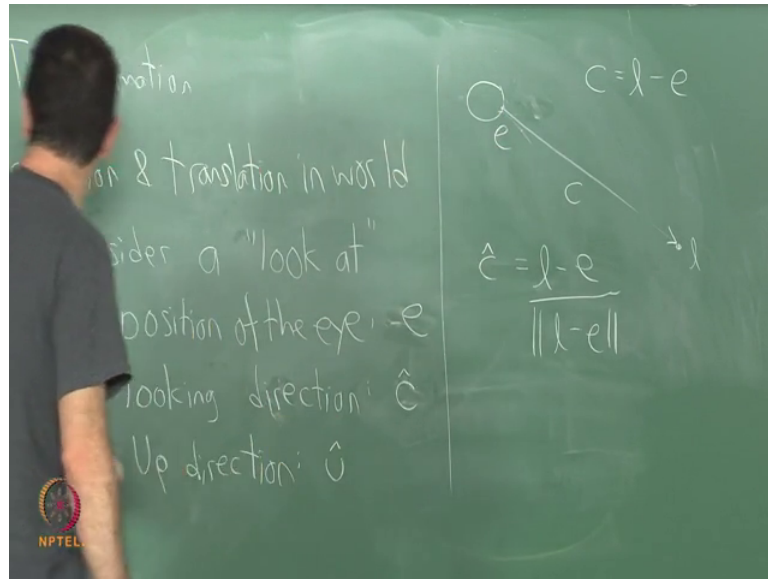


We have position of the eye, sometimes I will say position; sometimes, I will say translations, sometimes I will say orientations, sometimes I will say rotation.

So, rotation and translation are the operations and then the result is orientation and position correct. So, it is to interchange those as long as you understand what is going on. So, 2; I want to have a looking direction looking direction and three I need to specify an up direction. So, let me give some notation to these and I will explain it. So, the position of the eye I will just use as a vector e . So, I will just write it as e like this single letter and then the looking direction will be c ; a little hard to write here, I will put a hat on it, did you note that we have unit vectors and you put a hat on to notice a unit vector if you do not like my hats leave them off, but there are unit vectors. So, it is trying to denote it somehow, all right.

So, how can I obtain the look at here is one possible way to do it.

(Refer Slide Time: 05:50)



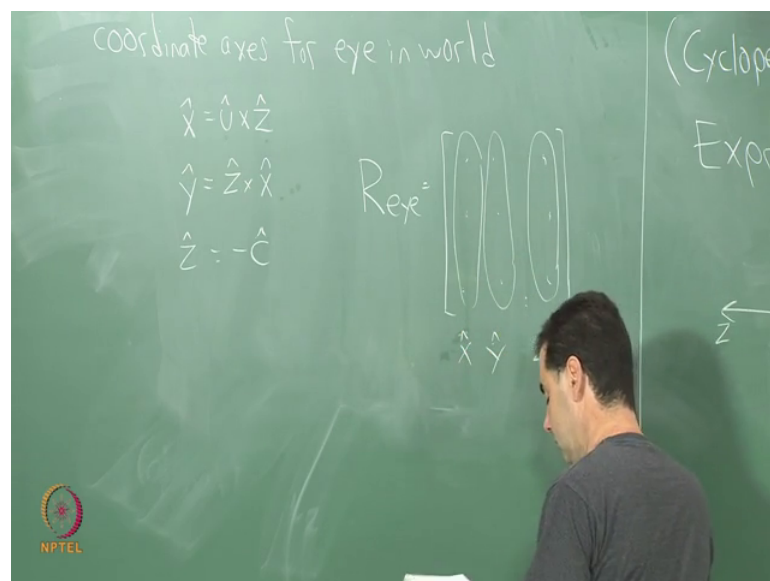
So, I have the eye and I have determined that the eye decided that the pupil is going to be located at some location e . So, it is very easy to write down. So, imagine you are given where the eye is, it will have some coordinates that correspond to where the pupil is at I will call that e and then if you are told that let us see if you are told that the pupil is looking at a particular point see you get rid of that.

So, the eye is pointed towards a particular point here call it l , then you get a vector, let us say a vector is that corresponds to traveling from the pupil to the point l . So, I can just call this vector c , I will say that c is just using the simple coordinates. So, you have here the Cartesian coordinates of the word l minus e , all right. So, that gives me that vector, right. So, it gives me that vector and then my c hat which is the looking direction here number 2 and I have given you number one by just simple specification number 2 I just normalize that vector.

So, l minus e divided by the length of that vector l minus e , correct; so, that is I am true what am I doing here; I am trying to specify the coordinate frame that I get here right how is it oriented in the world when the eye is looking at a particular place. So, I just want to describe that I want to figure out that coordinate frame and then from that very easily construct a rotation matrix that will put the eye in that orientation and of course, construct a full homogeneous transformation matrix that will also put it in the right location, but that part is considerably easier because it is just translation by e .

So, I need to figure out the orientation of this I am trying to put together the coordinate system it looks like once I figured out \hat{c} , I should essentially have the z power, right, it is just pointing in the wrong direction. So, it will be the \hat{c} corresponds to minus z and then there is also a possible role with respect to that that has to do with the up direction which we will have to take care of and once we have the up direction and we have let us say the forward backwards direction we can do a simple cross product right handed cross product to determine the third axis the third direction and we have agreed to do right handed coordinates and so, that will matter.

(Refer Slide Time: 08:41)



So, if that is the case, then if I want to figure out these axes. So, let us say coordinate axes for the eye in the world, then I get \hat{x} , \hat{y} , \hat{z} for these and as I said \hat{z} is going to be minus \hat{c} . So, I am figuring out that one, if I have this eyeball oriented somewhere I am figuring out that part and then \hat{x} ; \hat{x} I take the up direction and I cross it with \hat{z} , right. So, I take my up direction, I cross it with \hat{z} and I guess \hat{y} should be should \hat{y} just be \hat{u} that make natural sense.

So, when I started teaching this I thought why should be \hat{u} and then I noticed; it was not in the textbook; for example, by Shirley that we are using and it is not in a lot of the graphics literature and it is instead just \hat{z} cross \hat{x} and \hat{z} hat cross \hat{x} hat which is also fine the reason why they do not use \hat{u} here directly is just in case someone gave you \hat{u} that was not perfectly parallel to the image plane right. So, it does not correspond perfectly it

is not; so, if I look; if I am; if the eye is actually my eye, then if you instead of pointing straight up is slanted some this would correct for that if you do it in this way.

So, it will tolerate a sloppily defined up if you want. So, so yes.

Student: (Refer Time: 10:22) e and (Refer Time: 10:23).

These are all in the world frame correct yeah I am trying to imagine that the eye has been placed into the world frame and I am working with a description of that somebody tells me well the pupils at this location and it is looking at this particular point in other words that point should be in the center of its image right that is the point l here that would be in the center of the image from that, I make this vector c by just subtracting these 2 points.

And then I get the look at direction essentially right like what way are you looking and then the only thing that matters is you know is the camera upside down right side up how exactly is it oriented that is the next part that were trying to figure out which is handled by this last step, I am taking the up into account here and I just keep performing Cartesian products to get everything to get my coordinate frame orthogonal and right handed.

Regardless of whether somebody gave me a sloppy u value that is not perfectly orthogonal to my other axes; so, it allows you to be sloppy with you personally I would say get it right the first time do not be sloppy about it, but this will fix a little bit of slop in that. So, once I have that I get now a rotation matrix R eye where let me just write it out with dots and then these columns this column is just \hat{x} this column is \hat{y} and this column is \hat{z} .

So, this gives me a rotation matrix do we agree with that now this gives me a rotation matrix that if I start off with my eye defined in its body frame and I apply this rotation matrix to it that I have constructed it should it does not translate the eye, but it should rotate the eye. So, that it is looking in the right direction do you believe that that. So, that is what we have if I want to also translate it then I would after performing the rotation I add e the x , y and z components to the result.

I can do it all together in one homogeneous transform does this solve our problem of trying to figure out what the perspective of the world is from the eye from the eyes coordinate frame I think I did it backwards did not; I figured out from the coordinate frame of the world how the eye appears I need to do in the other direction how do I solve that alright. So, so let us see if I am you know I am standing here I turn my we talked about this before I turned my head let us say counterclockwise 45 degrees, it how does the world appear to have moved from the reference point of my eye if I hold my eyes still it appears that the world has rotated in the opposite direction right it is an inverse transform.

So, that is what we actually have to apply we do not want our eye we want the inverse and we do not want the translation e we want the inverse and so, the way to imagine it; it is as if your eyeball is being held fixed and it is the world that is moving around all the time quite wildly I guess because if I turn my head quickly it would take a lot of energy to move the world around that fast right. So, it is a bad relativity problem right, but that is that is how the mathematics feels like when you perform these transformations.

So, the eye is being held fixed all the time because the screens is being held fixed and relative to the eye and you are going be presenting images to that ultimately. So, this is being held fixed in the world is being transformed. So, we need the inverse now remember I just gave you a warning about taking the inverse of one of these transformation matrices. So, let me give you the inverse now and then we will stop and take a break which I seem to have forgotten to do I got a little too excited. So, let us see.

(Refer Slide Time: 14:15)

$$T_{eye} = \begin{bmatrix} \hat{x} & 0 & 1 & -e_x \\ \hat{y} & 0 & 0 & -e_y \\ \hat{z} & 0 & 0 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, let us give you the final result. So, T_{eye} is equal to well I need to have as the columns \hat{x} , \hat{y} and \hat{z} , sorry, I need to have as the rows, I said columns is wrong I need to have as the rows \hat{x} , \hat{y} , \hat{z} because I am performing the inverse transform also I need to extend this because I am not just going to apply rotation here I am going to put it together in homogeneous form.

So, this is just the rotation part this corresponds to that R_{eye} there transpose which because it is a rotation matrix this corresponds to R_{eye}^{-1} and then I have the remaining components and I make this. So, that undoes the rotation and then I have the undoing the translation part which should have my negative e coordinates here negative e_x negative e_y and negative e_z , right. So, I have inverted the placement of the t_{eye} I undo the translation which is this matrix here it is on the right. So, it gets applied first and then I undo the rotation which is the transpose of the rotation matrix that we build using to look at you do not have to build from a look at you may get it directly from a filtering algorithm that is doing head tracking, right.

So, it may come from another place, but traditionally in graphics it comes from a look at. So, I wanted to build it from that as well, but it may just come directly from them from the tracking and it may come from some combination of the 2 as you move a controller around. So, it will come from different sources; it is not a constant, but once it is given and fixed this is how you obtain $T_{sub\ eye}$; questions about that.

Student: (Refer Time: 16:14).

How do you find.

Student: (Refer Time: 16:18) are you find the rotational.

Oh, I think I just did it. So, I mean I basically if you remember when I was covering three d rotations. In fact, when I went all the way back to 2 d rotations I was trying to convince you that the columns correspond to how the coordinate axes get rotated right I did that in the 2 d example and a three d example was the same way that is why we had these constraints remember when I was saying the columns have to be orthogonal.

So, they have to have a inner product of 0 and they have to have length one. So, these are how the coordinate axes are transformed by the rotation. So, I figured out what these are by the geometry of the look at and then I just write these directly into the matrix and that automatically forms the rotation. So, it is a very good question and it is a very easy thing to forget, but whenever you look at a rotation matrix remember that the columns are these are the coordinate axes normalized and showing exactly the effect of this rotation matrix on the standard basis elements.

So, if you take the x, y, z basis elements that are $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and sorry I should start I could go right in order if you take the basic coordinate axes $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ transform to three of those with this rotation matrix, you will get exactly the expressions in these columns. So, it is just a direct step long answer for a short question hopefully its clear.