

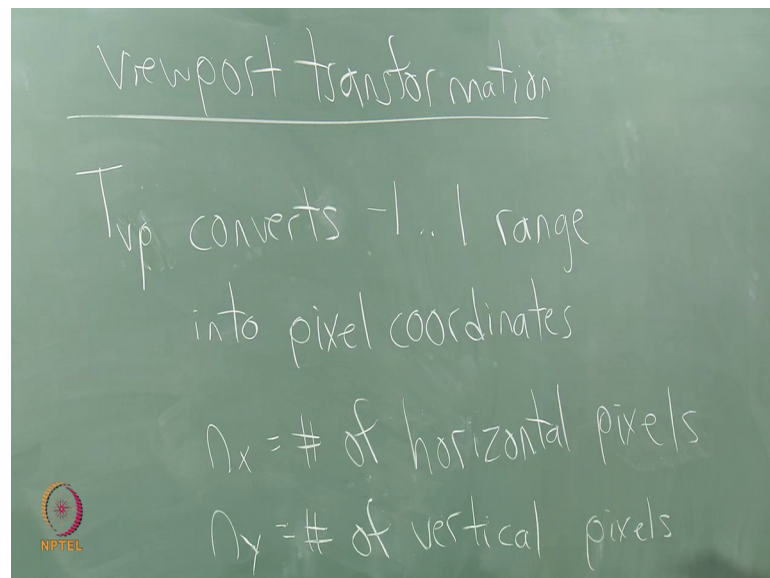
**Virtual Reality**  
**Prof. Steve Lavalle**  
**Department of General**  
**University of Illinois**

**Lecture – 6-2**  
**Geometry of Virtual Worlds (viewport transform)**

So, I have one more transform to add right what was the remaining one anybody remember any more viewport. So, again that ones like a very simple like redoing of units or something, we are going to go to pixel units. And at this point we do not even care so much about the z coordinate, it is assumed that the rendering engine performing a computer graphics operations has used the z information already for us.

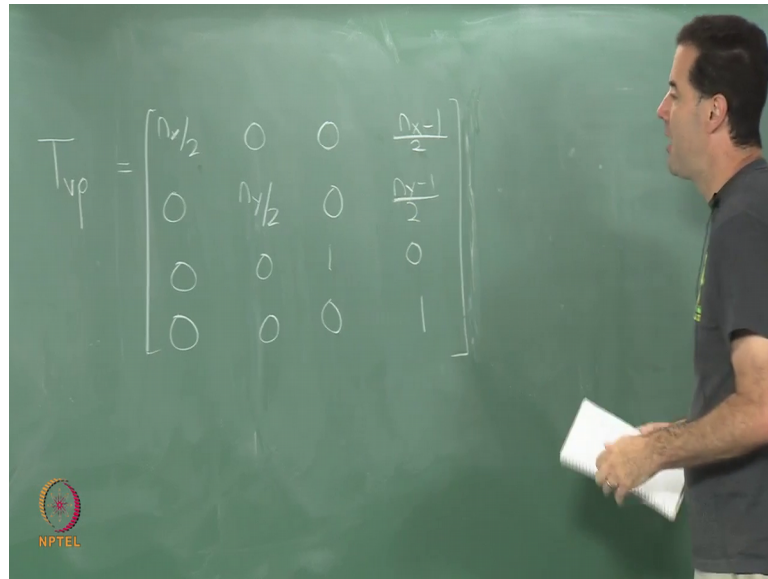
We just need to figure out now how to put the points in the coordinates of the screen. So, remember that the viewport.

(Refer Slide Time: 00:57)



So,  $T_{vp}$  converts the negative one to one range into pixel coordinates. So, let us say  $n_x$  is the number of horizontal pixels, and  $n_y$  is the number of vertical pixels. So, for example, in a 1080 p display  $n_x$  would be 1920 and  $n_y$  would be 1080.

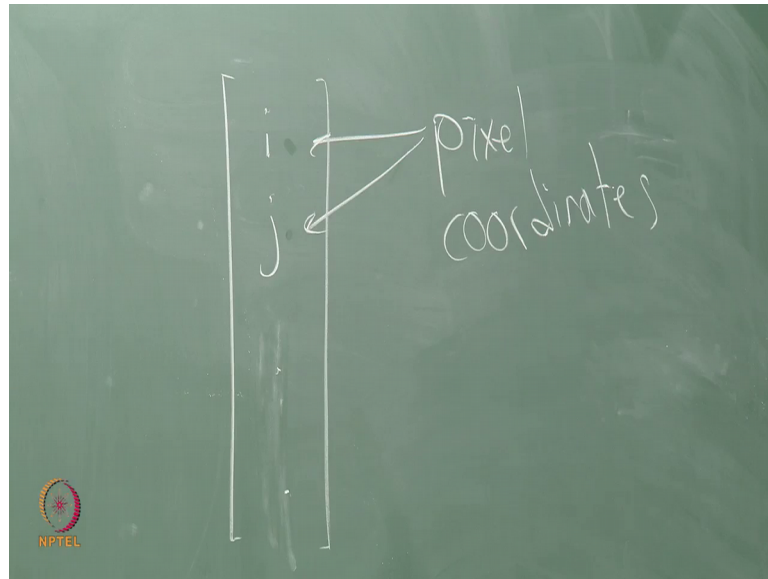
(Refer Slide Time: 02:22)



So, if we are going to go and write to that then for a simple example like this  $T_{vp}$  you can get to make a 4 by 4 matrix, and  $x$  divided by 2 0. So, I cannot divide that 2 out because I made a in a canonical box that we made right it was a the cube had width 2. So, one of the arguments for making it with one would get rid of this 2 here. So, you know I always tradeoffs in these different choices that you have again we will do a scaling here for  $y$  and  $y$  divided by 2 0 I do also do a displacement because 0 0 is usually not the center of the screen right 0 0 is the upper left corner or lower left corner depending on how things are written in your in your arrays how the array access is done.

So, I have to do some offsets here I have to shift it by roughly half which is  $n_x - 1$  over 2, and  $n_y - 1$  over 2. So, that is the translation part and the rest I just make it like the identity matrix, because I do not want to cause any harm right. So, if I apply this to some I should say if I apply this to everything else that comes before. So, if I use  $T_{vp}$  now and I apply this to result that I had before, from the entire chain of transformations after I get done applying  $T_{vp}$  the result will be some 4 D vector.

(Refer Slide Time: 04:07)



But I have the  $i$   $j$  pixel coordinates here and I do not really I do not really care about what is in these not really sure what to put there. I will put some dots there I do not really care what is in the third and fourth positions the important thing is that I get what are called pixel coordinates and so, I get pixel coordinates, which way have I assumed here have I assumed that the  $0\ 0$  point on the screen is in the lower left or upper left corner it is upper left right.

So, if you want to fix it for the other way, you got to change some things right you change the  $y$  scaling should have a negative one multiplied into it, and then you have to figure out how to correct for the offset here right. So, it is no problem, but shift it around it is very easy there to get visual feedback if you want to experiment and you know it change the transformations here in the very last step, but that is all you do for this and we finally, have by a long day finished all of these transformations questions about this.

Now, I write  $i$  and  $j$  there because these are supposed to look like integers, that correspond to the exact pixels, but keep in mind today we have not done anything about quantization. So, after doing all of these transformations I expect  $i$  and  $j$  if I do them in a computer and do all these calculations even though I might have started with integers for the locations of my objects in the more bodies in their initial body frames, after doing all of these transformations I am going end up with floating point numbers all the way over here in  $i$  and  $j$  right.

So, I am not rounding them off I am not doing anything with that yet, but ultimately in the graphics pipeline you are going have to deal with that based on the resolution of the display number of pixels per axis, you will have to determine how to draw slanted lines for example, and deal with the stair casing issue or aliasing issue. So, that part I am not talking about today was just how to put all of the information, your model in the appropriate place on the screen or on the display questions and.

So, think I am I am done with this part and next time we will start on something quite different which is the physics of light, and we will start talking about optics and how that part works and then we will get into the human eye as an optical system and then talk about the human body in the context of that, and in the context of virtual reality. So, we got up to the display today just the mathematical parts, now we are going to get into the physics part of what happen when the display is in front of your eye.

Thank you.