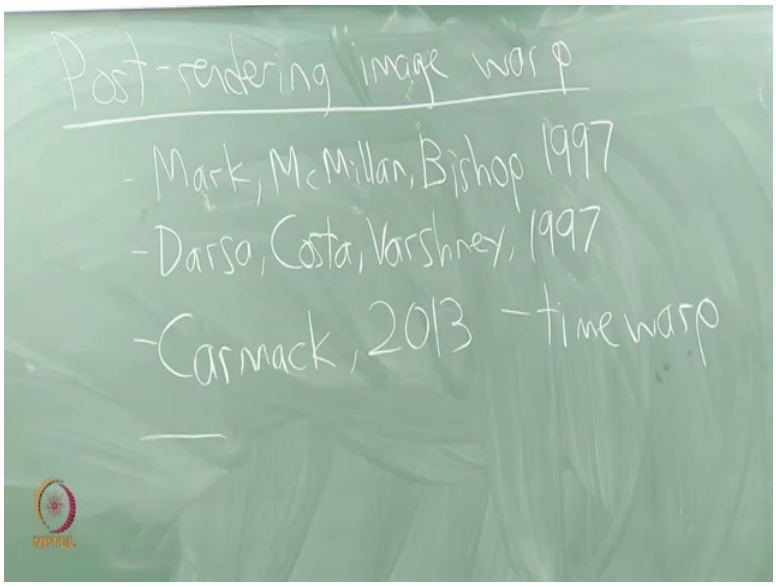
Virtual Reality
Prof. Steve Lavalle
Department of Multidisciplinary
Indian Institute of Technology, Madras

Lecture – 16 – 3
Geometry of Virtual Worlds (eye transforms, cont'd)

I want to talk about post rendering image warp as a technique to reduce the effect of latency in the rendering pipeline for virtual reality.

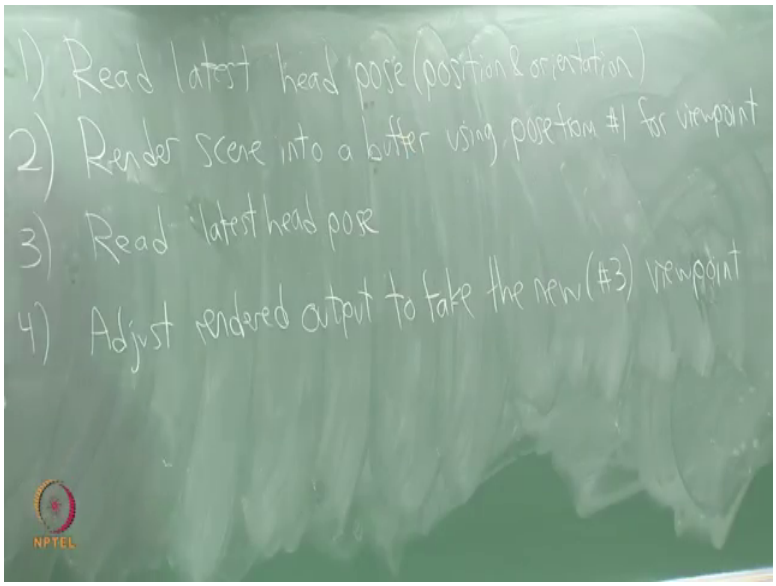
(Refer Slide Time: 00:30)



Post rendering image warp this was done by Mark, McMillan and Bishop in 1997 is also a similar work by Darsa, Costa and Varshney in 1997. It is also been reintroduced by Carmack in the industry in 2013 under the name time warp.

So, I will use the original name post rendering image work. So, see that around if you do a lot of searching on blogs and news media and things you will find a lot of talk about time warp as well. So, there are four steps to this.

(Refer Slide Time: 01:57)



One you read the latest head pose I pose I mean position and orientation also you query the tracking system and you ask which way is the head looking you get the most updated information and then you do the rendering parts that we have talked about. So, you render the scene into instead of putting it on the display you are just going to render it into a buffer. So, be like a kind of virtual display over to the side. So, render it into a buffer using this pose that you just got from a step one for the viewpoint.

So, far I have not done anything unusual that is what you should be doing, right. You grab your latest viewpoint and you grab the latest pose that gives you the viewpoint for rendering you do all of that, but you render into a buffer instead of putting it out onto the display and then after you have done this and it is assumed here that number 2 is going to take a lot of time and so, suppose this takes a significant amount of time hopefully it does not take a 100 milliseconds could in the worst case right I mean being well the worst case could be worse than that, but in a bad case it could take 100 milliseconds. So, let us say suppose it takes a lot of time.

Now, when it is finished what I would like to do is read the latest head pose, right. So, when I get up to step 3, I am kind of thinking why I wish I have this information back then, right because I just did all this rendering work and now I realize all the head somewhere else now. Now, again one of the solutions would be back here to just predict what you would see here. If you could perfectly predict what you would read in step 3

then you be in great shape that is what you should have used back here. You can try to predict some of this, but it might be too much also you might not be completely sure when step 2 is going to finish.

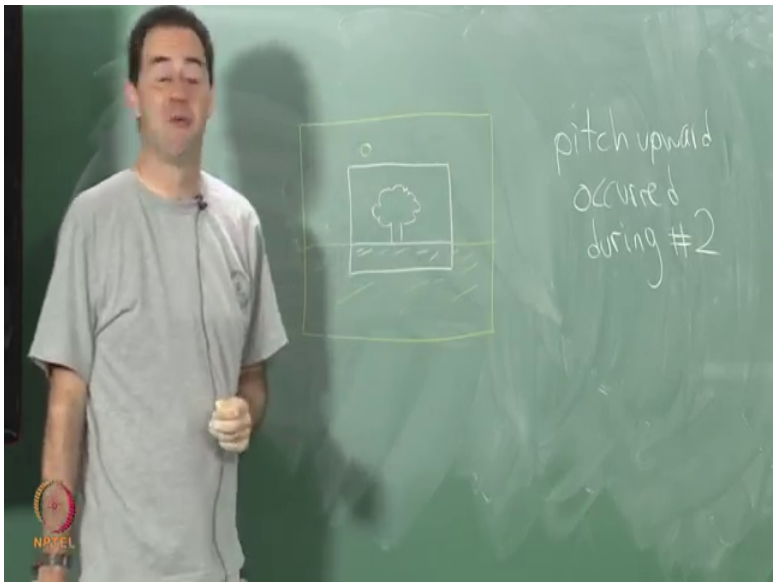
So, if you could estimate exactly when it is going to finish and based on the momentum of your head predict exactly where you are going to be then you should just use the pose you would have gotten in step 3 all the way back here just estimate it and use it, but suppose you can, suppose step 2 takes too long not quite sure when it is going to finish I read the latest head pose and then in step 4 you adjust in other words hack in some way the rendered output in that buffer that I that I stored it in to fake the new viewpoint. In other words, the viewpoint from step 3 viewpoint right.

So, what do you think about that. Let me give it a little bit of an example of that and then I will discuss some of the shortcomings of it, but when it works it is great. It might not work well and then it leads to some perceptual artifacts just to get you kind of ready for this. One of the things I could do is I could in rendering let us just go back to the stereo problem before talking about latency.

So, let us just say I want to render to each of my eyes I could do all the rendering for a Cyclops, remember we talked about the cyclopean view, so, I could just render for a cyclops and then just shift the images that I rendered by a little bit to account for left and right eye views, right, that might work. It is a trick you could do and that would save you from doing twice the rendering work and cut it in half which is a very big savings, but it might not be completely right, if my right eye can see the site of an object, but my left eye cannot based on where it is located. So, there is no way to fill in that missing information.

So, we can do some kind of cheats here to try to shift the viewpoint in some way that tries to compensate, but if there is some missing information then it is not going to work. So, let me give you some examples of how this looks.

(Refer Slide Time: 06:42)



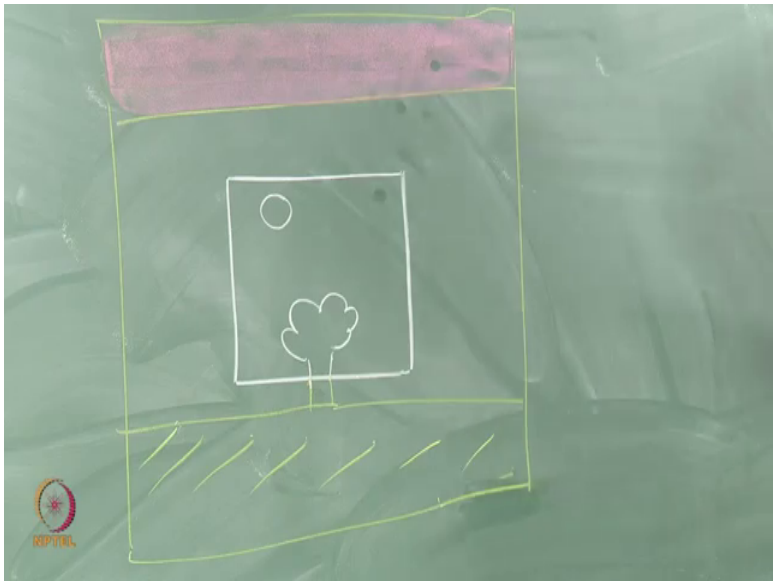
So, suppose this might be the screen that I was originally going to render to and you know have a nice image here let us say I put a tree here something some ground what I instead do is I extend this buffer that I am rendering to be significantly larger and I render more out here, right.

So, I may render this much more and now, after I get a new viewpoint I may perform some kind of shift to this I have all this extra information around that I can use. So, that I just shift the image that what I show on the white area is what gets rendered to the display, but I am going to use other information that I have rendered out here, how much larger this needs to be depends on how long it took during step number 2 here and how fast a head puts moving, right.

So, based on some simple bounds this might be just a thin rim around the edges or this could be enormous in which case if this yellow part is too large then you have rendered way too much, right. You have done too much work and it might not be efficient to do it this way. So, what if you have noticed that while you are doing the rendering a pitch upward occurred during step number 2 of the steps that I just erased, right? So, a pitch upward occurred during step number 2, right. So, this is how the scene looked we started doing the rendering right this is the view point all the sudden the person pitched their head up like that that should make this appear to move down right. So, all we have to do

to fix that is just perform a downward shift of the pixels if that is all if it were a pure pitch I think the right way to draw this here.

(Refer Slide Time: 09:02)

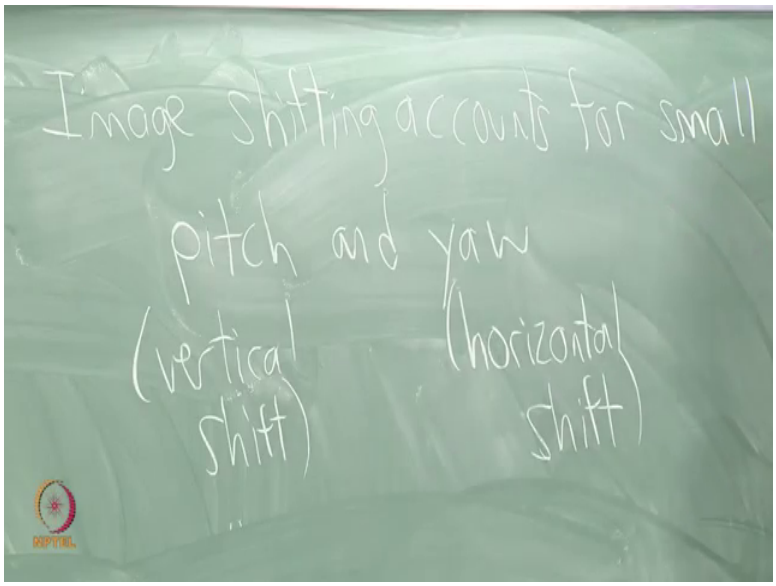


So, I have my maybe I just make it look like the white box went up, let us see how to undo this. Now, let us keep the white box in the middle here, let me see your just make sure it consistent ok, suppose this white box is in the same general place, but I am just going to shift what I have rendered downward. So, then all of a sudden I may just see the top of the tree here and then the rest is falling below and the ground is down here, now right and so whatever else I rendered here I took some of the part that I rendered up here and I just shifted it down and this is the part inside of white that I will be outputting.

So, whatever was up here maybe for example, maybe there is a let us say the sun is up here whatever some disc is up here. So, perhaps that got shifted down. So, that that does show up and get rendered on the screen, right. Now, what I do not have is up here I have some place where I never all that looks too much like my ground I should be careful let me put it a different color. So, I have this area up here where it is a kind of danger zone if I shifted too far down here, I do not have any information up here, right. I did not render up there.

So, if I had to shift too far I would be in trouble and all of a sudden I may have black or some other patterns showing up that I did not actually draw, right. So, I had be beyond my limits. So, I just wanted to point out that this happens when we do the shift, all right.

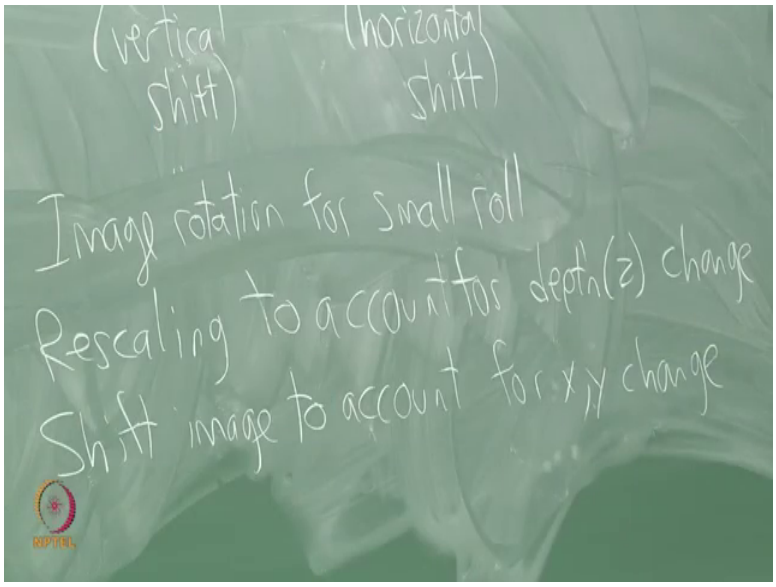
(Refer Slide Time: 10:55)



So, these image shifting these image shiftings accounts for small pitch and yaws right by it for pitch I can do a vertical shift and for yaw I do a horizontal then, right. So, I could do it in a horizontal way to account for a change in yaw what if there is a roll we could do a rotation, right.

So, we have to make sure we have enough information to do that; we should be able to account for small rotation. So, very good, we do image shifting.

(Refer Slide Time: 11:54)

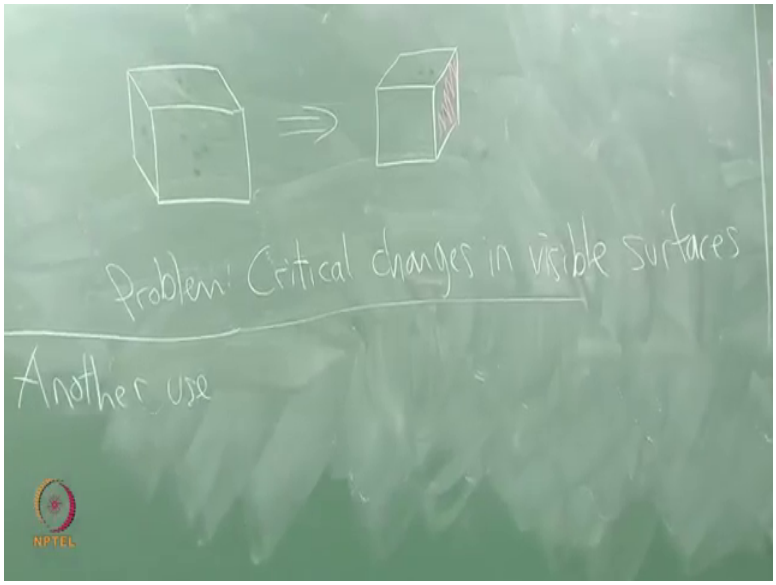


For that we do image rotation for small role. So, if there is a roll happening we do rotation how many more degrees of freedom do we have? That accounts for rotations what about translations that occurred right. So, what if while the rendering is happening I decided to move my head like this and come closer you did is zoom in, right. So, I guess that should work you can just zoom in on this not too bad.

So, rescaling to account for a depth change or z change so that might not be too bad I hope I have enough resolution to do that if I have to move inward, right maybe I messed up some you know maybe may have made a worse aliasing problem I doing that. So, we can do that what if I move side to side. So, that should be very so, should be more shifting I guess right side to side. So, more shifting, so, shift image to account for x, y change alright. So, that is all fine, but these transformations are not correct.

So, there is a great perceptual question is there any side effect of that is it uncomfortable is a disorienting is it unbelievable does it not look right, you know these are experiments that need to be done with human a subjects trying it.

(Refer Slide Time: 13:55)

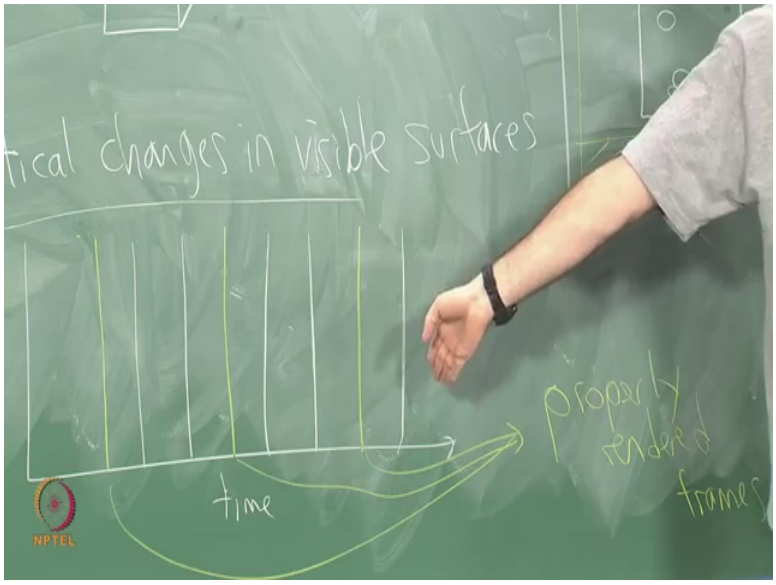


And, here is one of the fundamental flaws as I as I mentioned earlier when I was talking about stereo see. So, if I have let us say some cube it might be the case that I originally render these three sides of the cube, but then after performing an x shift I realized that the perspective should look like this, right and so, it may be the case that when I moved from here over to here I now see this side of the cube this side disappears. So, me side

that disappears fine, but look at this the entire shape is actually changing in some way to of this top here and there is this zone here where I have no idea what to render their right that into any of the work. So, you are in trouble.

So, if there are critical changes in visibility. So, problem critical changes in the visible surfaces or triangles in particular then we are in trouble we do not have we do not even have the right information. So, that is a fundamental difficulty nevertheless this techniques being used a lot in actual practical systems that are deployed and one additional thing you can do with this technique just have one particular final use of this.

(Refer Slide Time: 15:47)



So, final comment to make for this lecture another use of this is it might be the case that I have such a slow system that let me draw out a timeline here it may be that I cannot get a full frame rate. So, I can only generate a frame at this time and a frame at this time and a frame at this time and maybe I can only generate about 20 frames a second, all right. One thing I could do is I could use this these image shifting ideas to make interpolated frames could put a frame in here and a frame in here and a frame in here and a frame in here and so on. So, these yellow ones here that I have made are properly rendered frames, but the white ones that I have that I have inserted are just done by interpolating the viewpoints, right. So, I can just do some image shifting that corresponds to interpolating the viewpoints and that will fake more frames than actually have and that is significantly more comfortable.

So, for example, in some of the lab computers you may have difficulty with demos running at 30 frames a second if we could hack those demos. So, that we can insert additional frames we could get it up to the full 75 hurts that is required there by just putting in more frames I just have interpolated viewpoints and the graphics cards should be able to easily handle that because it is only on only performing a simple shift before generating the next frame we get the frame rate up it is not quite, right but it should be much more comfortable than letting it stutter or have this difficulty when it goes from frame to frame of a having the frame rate be very low. Does that make sense?

So, you can make interpolated frames by interpolating between the viewpoints and performing these image shifting techniques assuming you do not get into this kind of trouble that we talked about up here where you just do not have the information or there are certain geometric distortions like the top of this cube that are just not correct and there is an fascinating question which is where does perception play into this, right?

You can perform these tricks are they perceptible or not, right do they can you detect the problem or worse yet maybe you cannot detect the problems, but they might make it uncomfortable in some way right it may eventually contribute to fatigue or even nausea perhaps I doubt it for this particular case, but maybe it goes. So, far as to contributing to that one has to do significant human subjects testing to determine whether you know what the effects are of this in terms of perception and their effect on the human body; so any questions about this? Alright.

So, that is all I want to do to cover about this topic and for the next lecture, I will cover on the audio component of virtual reality.

Thanks.