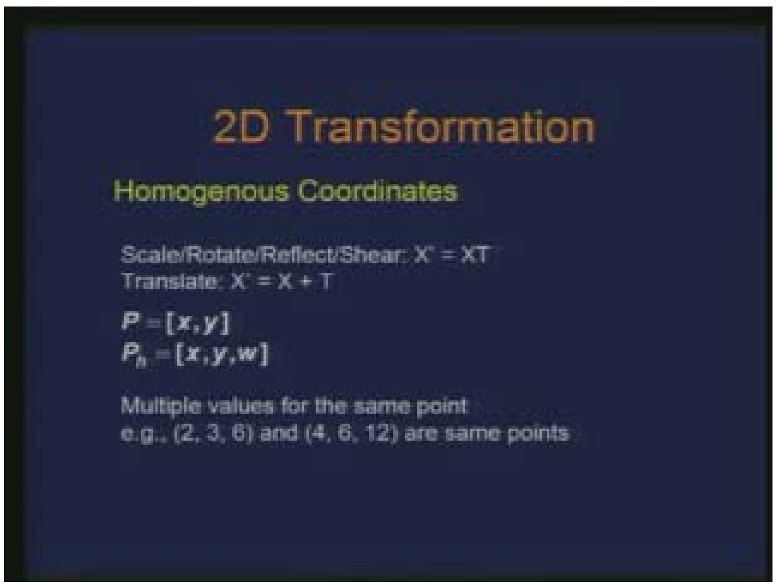
Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 7
Transformations (Contd...)

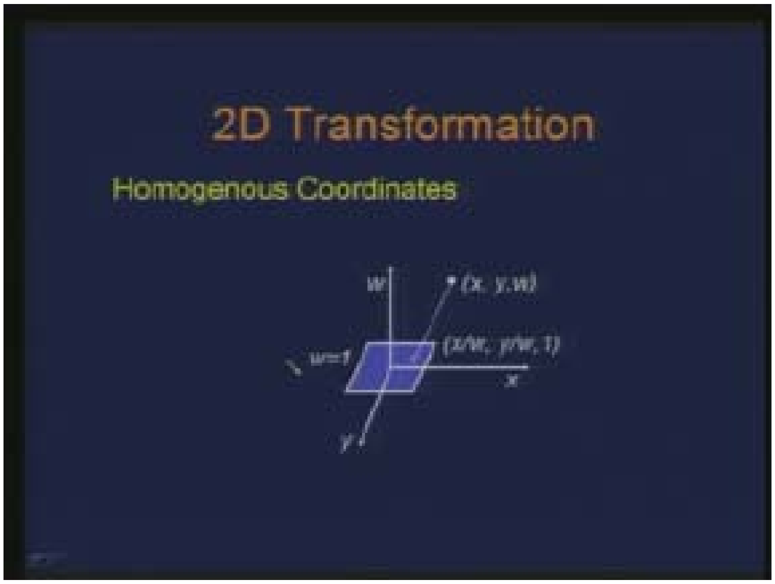
Last time we started talking about transformation, primarily the 2D transformations. So let me recapitulate some of the points which we talked about last time particularly dealing with the homogeneous coordinates. One of the motivations which we observed was to be able to represent the transformation in the form of a matrix and have the uniform or unified representation for all the transformations.

(Refer Slide Time: 1:19)



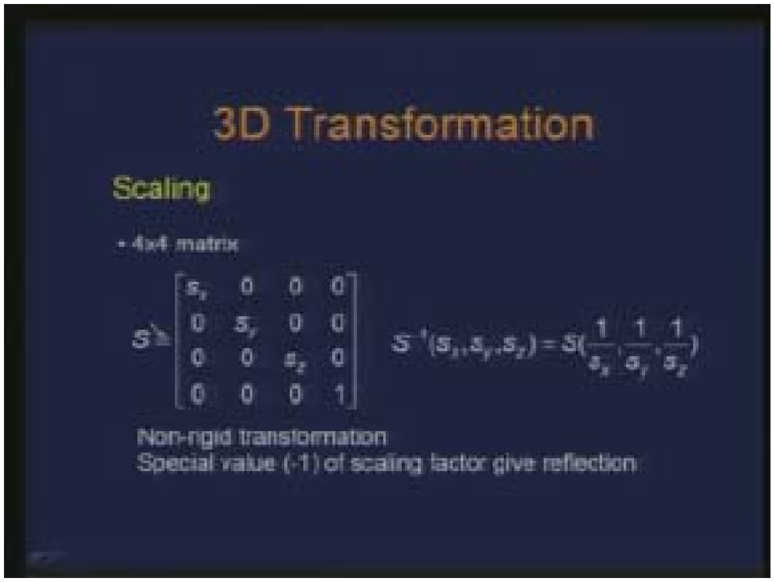
So looking at the nature of the transformation for instance in translation and the other transformation like scaling and rotations we wanted to combine these. Translation is the one which is additive and rotations and scaling and other linear transformations are multipliable. So we wanted to put it in the uniform framework. For that we introduced homogenous coordinates. And the idea there was we have these transformations like scale, rotate, reflect, shear being captured in the transformation T as a matrix and then we multiply that matrix to this x. Whereas in the case of translation when we have the offset of the translation vector given as T then we need to add that to the point X to get the X value. Now what we do is we basically introduce homogeneous coordinates where we add an additional coordinate which is W. We have the same point now P being represented as x, y, w instead of x, y. So, in turn what we observe that there could be multiple values for the same point. So what are we actually trying to say here is that we take the representation of the point which was in 2D space into another space with having a third dimension of w.

(Refer Slide Time: 3:29)



And in fact geometrically when we observe we get something like this, a space where we have a representation of the point in terms of a ray so we take the 2D space to a 3D space which actually has become the projection space because depending on what w I choose I get a particular value for the point which is the same point in the homogenous coordinate. So that was the symmetrical interpretation. So we basically take the representation of a point in a projective space. There is an interesting observation here, what happens when w becomes 0? When I have w is equal to 0 the point which I am talking in my Cartesian space is a point at infinity. So I have a mechanism now to be able to represent even points at infinity which I did not have otherwise in the Cartesian space. And these become more relevant when we actually talk about projections and perspective transformations because there we will be talking about the vanishing points. Vanishing points are actually points which are located at infinity, so how do we represent those points? This basically facilitates us to be able to represent even the points at infinity just by giving value of w as 0. Then we also looked at 3D transformations. So, 3D transformations again in the homogeneous coordinate system can be represented through a 4 into 4 matrix. Let us look at scaling.

(Refer Slide Time: 6:05)

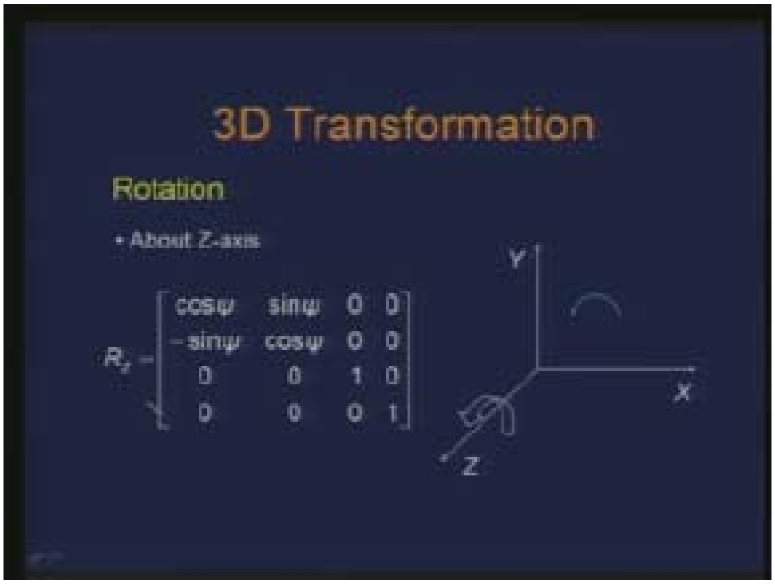


Scaling is now a 4 into 4 matrix where the diagonals contain the various scaling factors such as S_x S_y S_z. We know that scaling is actually a non rigid transformation. Non rigid transformation means that the shape and the size changes. There is a distortion of shape and there is distortion of size particularly the size, the shape as such remains the same but the size changes. Even when I have the non uniform factors of scaling the shape still changes. These are basically non rigid transformation because I am not able to preserve the shape. And an interesting thing is that when I look at the inverse of the scaling matrix which is represented as S I can construct that inverse transformation just by taking the reciprocal of the scaling factors. The matrix consisting of this actually gives me the inverse transformation of this scaling because at times we require to perform an inverse operation of scaling so there it is very easy to determine the inverse transformation.

And some special values like if I have the scaling factors negative then also I have the reflection then we look at the rotation. So the rotation matrix now in fact it is basically the same extension what we had in 2D rotation. The 2D rotation implicitly involves the axis coming out of the plane where the rotation is taking place. Therefore rotation is actually about an axis. When we look at rotation in 3D then we consider the various axes about which we can perform the rotations and the matrices we obtain from those rotations.

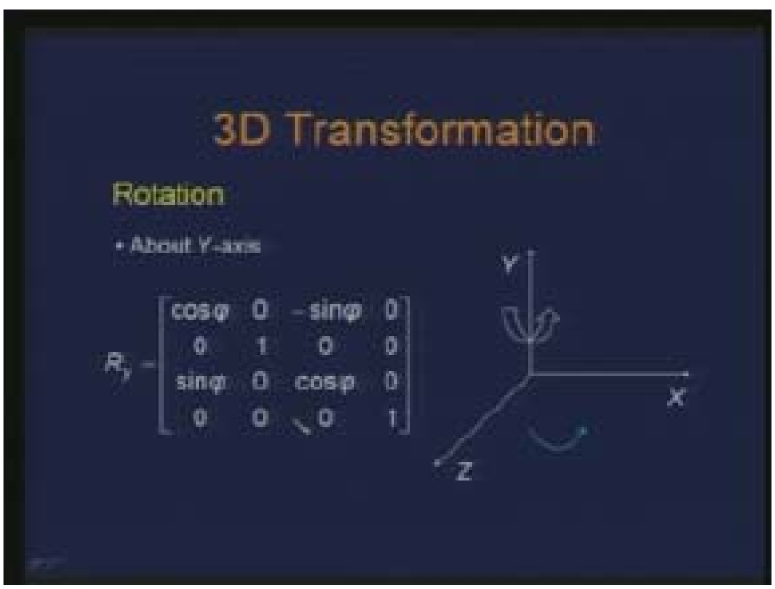
For instance, if I am interested in rotation about x axis, and in fact you can look at something like a rotation which would have been performed in this YZ plane. And the rotation would have been just this sub matrix here. And remember we have the convention of positive rotation in a right handed coordinate system given by the orientation of the fingers when the thumb is representing the axis of proportion so we keep that convention in mind.

(Refer Slide Time: 9:43)



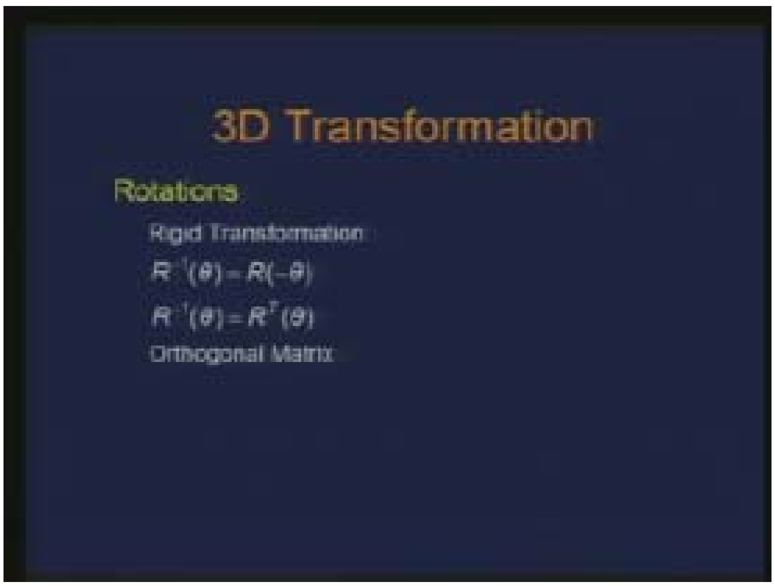
Now if I look at for instance rotation about the z axis this is what the scenario is. So again I can look at this rotation as something similar to as if there is a rotation in XY plane which is given by this. So I get rotation about Z axis. If I was just to do the rotation in 2D then I just reduce this matrix cutting down this row for Z and this column. So basically I get a 3 into 3 matrix.

(Refer Slide Time: 10:33)



Now when we talk about rotation about y axis this is the scenario. Again I can think this to be a rotation being performed in XZ plane and remember we have this convention of positive rotation. Now when I perform in this rotation in XZ plane you see that the rotation is happening in this direction and often we measure the angle in this direction. In effect we are performing the rotation by the negative angle and that is why this minus has now come here. In the earlier rotations you had seen that the minus was here so this is a positive sign. Now there is a flip of it because basically I am preserving my convention of positive rotation and therefore this rotation is in effect by in an angle minus pi. Now the rotation is happening in this direction that is my positive rotation because then I would have the y axis acting as a thumb. If I were to measure the angle from x to z what would have happened which would give me the rotation? There the y axis would have acted in downward direction so I just reversed that. Therefore the changes sign occurs.

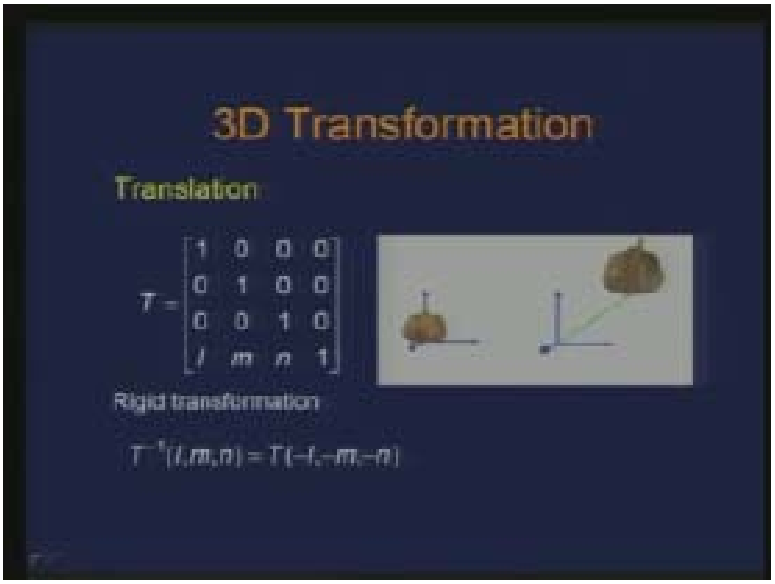
(Refer Slide Time: 12:39)



Let us say if I am taking about rotations in general. It could be rotation about x axis, it could be rotation about y axis, or it could be rotation about z axis so it is a class of rotation transformation, basically they are rigid transformation because there is no change in shape or size.

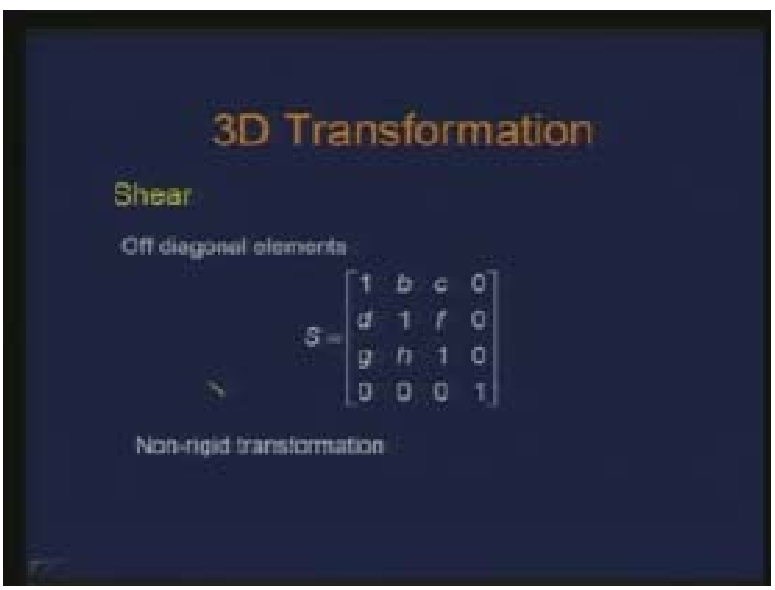
Another interesting observation is that if I have to invert my rotation therefore I am trying to find out the inverse transformation for the rotation then all I have to do is perform a rotation by the negative angular rotation and that is easy to verify. Similarly, I also observe that the inverse matrix for a rotation by an angle theta is actually the transport matrix of rotation by the angle theta. Therefore when I am trying to find out the inverse of a transformation all I have to do is just transpose that matrix. And in fact that gives me that the matrix is basically an orthogonal matrix. So what can you say about the determinant of this matrix? The determinant is unity. So orthogonally you know what that means. In fact you can consider the column of a matrix and then you can check that orthogonality using a dot product of that column matrix. So these are some properties of the transformation and the corresponding matrix of a transformation.

(Refer Slide Time: 15:24)



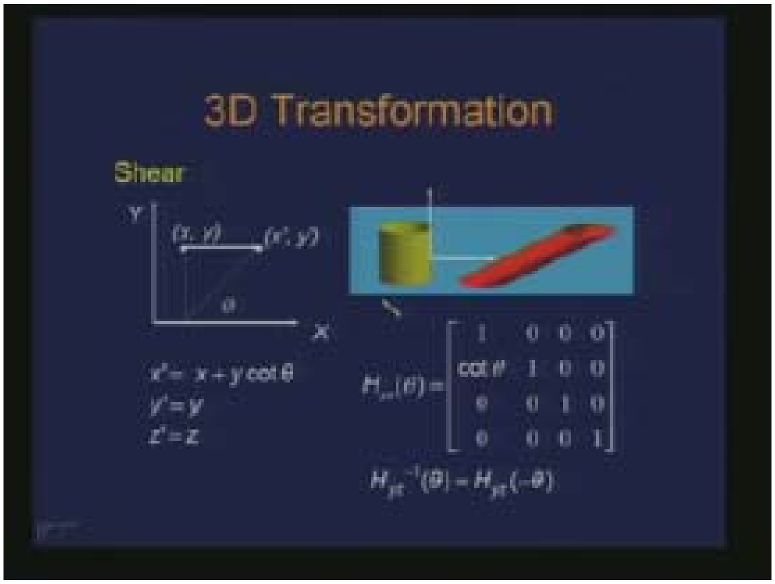
Now, similarly for translation when I am to perform the translation basically an object is translated to this location by an offset vector. Then again translation happens to be a rigid transformation and the inverse of the translation matrix can be obtained just by taking the negative offset. So l, m, n which was the offset of translation now if I use just use as minus l, minus m, minus n in the same matrix it gives me the inverse of the matrix of translation using l and m as the offset.

(Refer Slide Time: 16:19)



Again when we see the transformation of shear basically here we have the non 0 off diagonal terms that give us the shear transformation and shear is a non rigid transformation we observe a distortion in the shape. Now just in a similar way as we had seen for other transformations how we find out the inverse of this matrix course in general we need to find out the inverse as a general matrix. But for special cases we can again find out the inverse as a simple transformation in the values of the transformations.

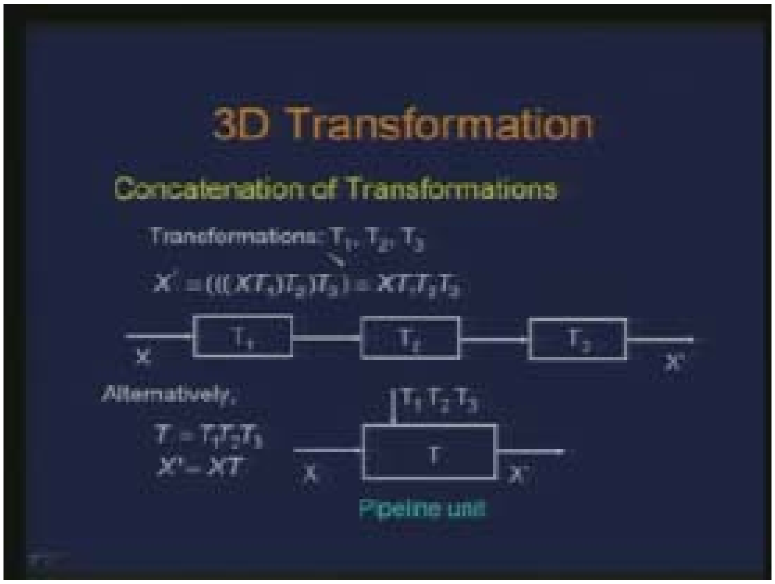
(Refer Slide Time: 17:13)



Here is an example where we consider a simple shear where the object is basically sheared in X. If I just look at the plane XY and I assume that there is a no change in the Z values then the point XY is changed to X prime Y prime. Now if I see what X prime Y prime becomes and I assume that there is this angle which is made by this line so I take the projection of this point on the X axis and this is the angle which is made by this X prime and Y prime from the X axis theta. Then I get X prime is equal to X plus Y cos theta Y prime is equal to Y and Z prime is equal to Z. So there is no change in Y and Z values. The only change is in the X value. So in order to have this transformation this axis should be strongly placed. In order that I obtain this where this is my coordinate axis having X and Y. This should be actually located at the center of those because it is a symmetry then the transformation is a symmetry point.

Now, using this as a transformation I construct my matrix of transformation for shear where I get this term of cos theta here. Now for such a type of a shear transformation and I call this as shear YZ because I am not changing YZ so there could be convention saying that this actually shear in X and there could also be conventions saying that there is no change in YZ so let us say I have the convention that when there is no change in that coordinate I call this to be H_{yz} and there is no change in YZ. Now if I have to take the inverse of this, this is equivalent to just change the sign of the angle. Once again the inverse of this is obtained just by changing the sign of the angle theta. But in general if I have a shear and mix of shear then I may not be able to do this direct substitution of sign of a value. Hence then I may have to do an inverse of a matrix as I would do for a general matrix. As we saw the composition of transformation in 2D the same concept extends here.

(Refer Slide Time: 21:08)

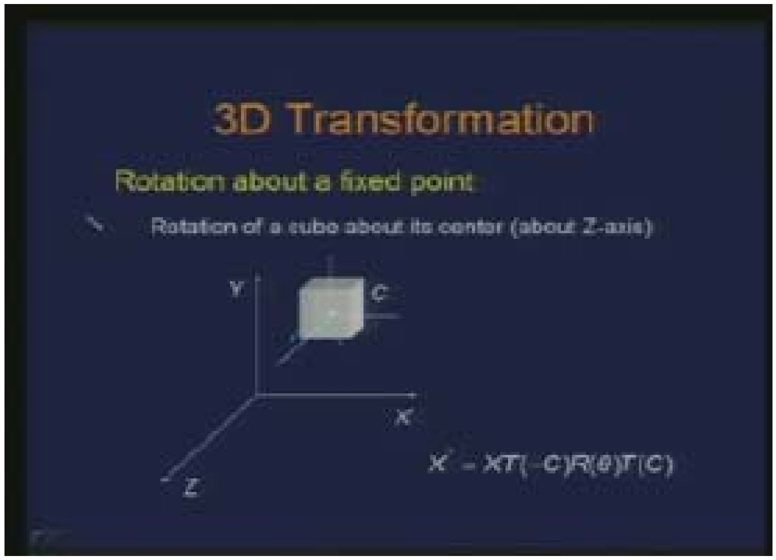


Now let us say I have a sequence of transformations given as T_1, T_2 and T_3 there could even be more but I just have these three transformations to perform. Then the transformed value X prime is basically T_1 applied to X and whatever I obtain I apply T_2 on it and then whatever I obtain I apply T_3 on it. This I can write as simple multiplication of T_1, T_2 and T_3 because I have law of associativity. So I can just write like multiplication of matrices T_1, T_2, T_3 to X. So, in some sort of a pipeline architecture I can think this to be X given as an input and I have these pipeline units where this units contain a transformation T_1 then the result goes to the other unit where I apply the transformation T_2 then the result goes to the other unit when I apply the transformation T_3 and I obtain the result X prime.

Here if I have an object represented by some hundred points or maybe thousands of points so what I will be doing? I will be taking them point by point and apply these matrices to this point in this pipeline fashion where each unit is actually a transformation. Now there is a problem in this formulation in terms of computation I am performing. So alternatively what I can do is actually combine all these transformations just by concatenating these matrices of transformation and I obtain the final transformation which is the concatenated form of all the transformation as T and I just apply that T to the given point X.

So this pipeline is actually now changed to a structure like this where I have an input X and the pipeline unit contains the concatenated or the multiplied transformation T_1, T_2, T_3 and I obtain X prime. Therefore here once I have done this multiplication of matrices I just need to apply one single matrix to all the parts which is computationally beneficiary.

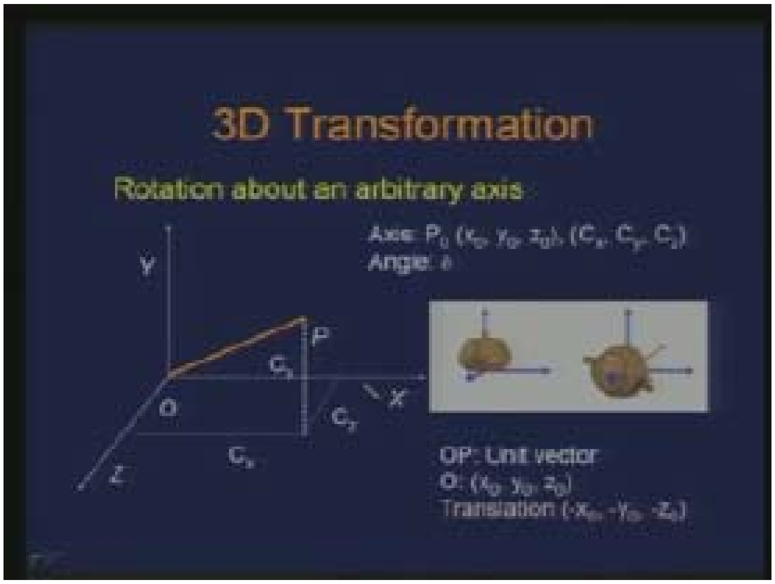
(Refer Slide Time: 25:05)



Again the way we had seen in 2D composition of transformations can also be illustrated in 3D if I take an example let where I have to perform rotation about a fixed point. Normally we have seen the point about which the rotation is happening is the original. Now I want the rotation to happen about a fixed point which is not to worry. So let us say this point is the center of the object these cuboids and I have to perform rotation about Z axis. Therefore I am again considering the rotations to be about one of the coordinate axis X Y Z. So, if I have to do this then I just have to make this point to be the origin and in order that I do this I need a translation. So once I do the translation of this point C to the origin here and then I perform my regular rotation and then I take this point back to the point where it was then I am done. Therefore that is what is done here, we have the three transformations, one translation which is taking the offset which brings this C to the origin therefore I have the offset given by minus C rotation by angle theta and then again I translate it back.

Here is a more complex scenario where I would like to have the rotation about an arbitrary axis. So far we have seen rotation about the coordinate axis. Let us say I want to perform rotation about an arbitrary axis. So how would I do it? Remember that since we know the rotation about the coordinate axis we need to map the problem to one of the coordinate axis. The basic thing is that I need to align the arbitrary axis to one of the coordinate axis then I can perform the regular rotation and then I do the necessary inverse transformation to put that axis back that is the basic idea.

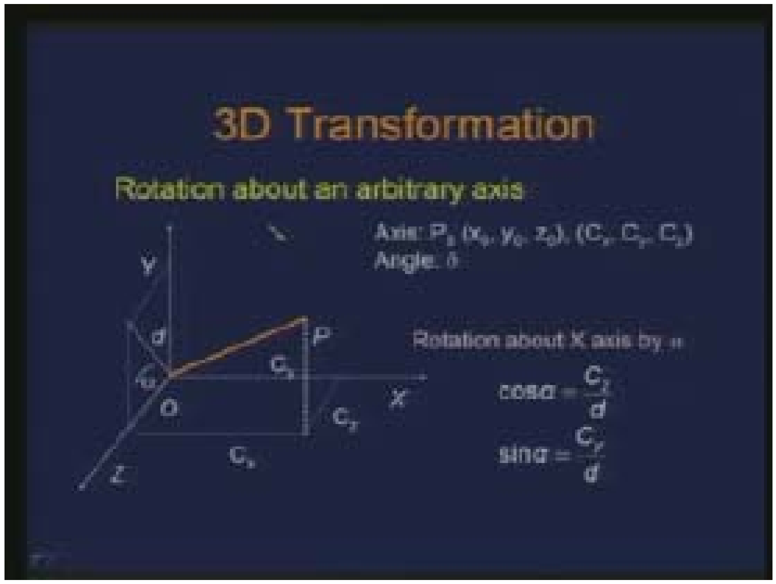
(Refer Slide Time: 28:34)



Here I am given the axis by the point P_0 so the axis is passing from a point P_0 given as X_0 Y_0 Z_0 and its direction cosines C_x, C_y, C_z so these are basically the direction vector of the axis and I want to rotate about this axis by an angle delta. So this is the statement of the problem. So here there is a small illustration where a rotation is basically being performed about an axis given by this red line and you get a rotated [k….] 29:28 like this. Now what I can actually do is first of all I can make this point the origin and that is the first thing I can do that is easy, I just need to make this point the origin which basically requires a translation by minus x_0 minus y_0 minus z_0. So what I have obtained now is this axis here passing through O and specified by this unit vector OP whether direction of this unit vector coincides with what I have here from the direction cosines of the axis.

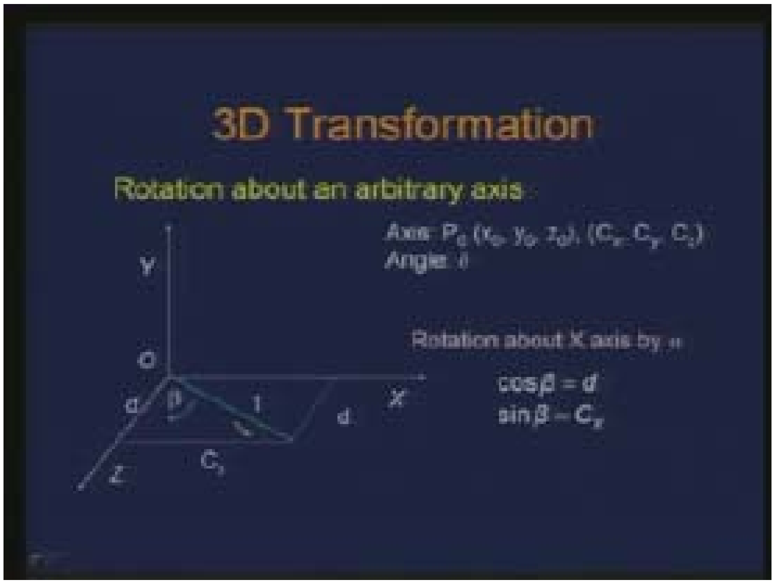
Now if I have to rotate about this arbitrary axis OP we said that we would like to basically align this to one of the axis and I choose this to be the Z axis so I want to align this OP vector to Z axis so how can I do that? Will that give you the alignment? First of all try to identify what transformations will be required to do this. Rotations are required. Now what we do is we perform rotations align this line to Z axis. Therefore as a first step what I do is I get this line in XZ plane which basically mean I rotate about X axis. If I have to do this rotation let us try to see the values of angles by which you have to do the rotation so that we can form our matrix.

(Refer Slide Time: 32:31)



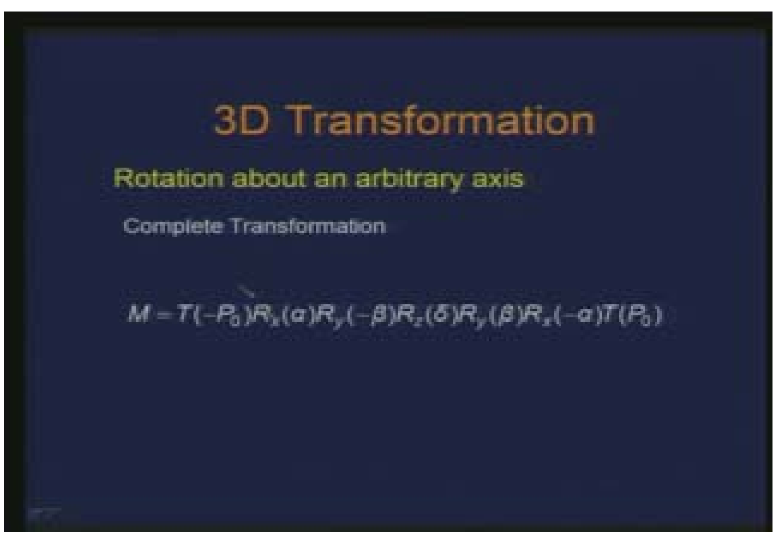
Now if you see this line in the YZ plane you would find this line being a sort of a projection of this on to the YZ plane given by d where d is the length of this line. And now if I perform the rotation of this line by the angle alpha the angle can be measured in this plane now. So, if I perform a rotation of this line by an angle alpha about X axis I would put this in the XZ plane. So when I am performing the rotation by the angle alpha the alpha basically can be computed through here in the YZ plane because I have this as Cz and this as Cy and this is d so cos alpha is actually Cz by d and sin alpha is Cy by d and these are the terms I need for performing a rotation. I need not explicitly compute the angle but all I need to do is get the relevant terms which I require in the rotation matrix. So I get cos alpha and sine alpha from here. Now as a consequence this axis OP has actually come on XZ plane. Now another rotation of that about Y axis would give me the necessary alignment.

(Refer Slide Time: 34:54)



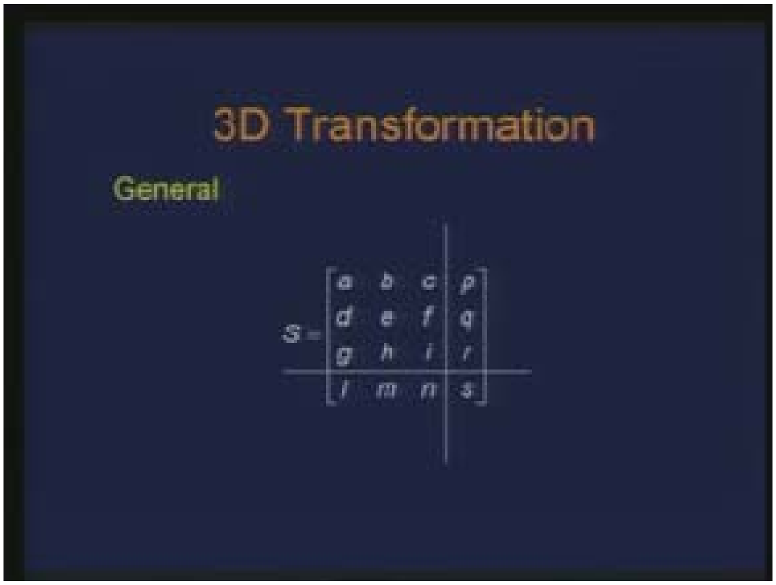
This is the axis in the XZ plane, remember it is a unit vector and that is why I have a length as 1. Now this side is nothing but d because when I perform a rotation about x axis if we go back here this d will actually come on the Z axis. So I have this distance given as d and now if I perform the rotation about Y by an angle beta I will align this to the Z axis. Now again we compute the terms which are necessary for the computation of rotation matrix which are cos beta and sin beta so cos beta is nothing but d and sin beta is nothing but Cx. So I have basically obtained all the necessary terms which I require to perform these rotations. Once I have done the alignment to Z axis then I can do rotation about Z axis for which I know what transformation to use and then I do the necessary inverse operations to put the axis back.

(Refer Slide Time: 37:03)



Therefore the whole pipeline of transformation would basically look as something like this. I have a first translation then I have the rotation about x by an angle alpha then I have the rotation about y minus beta again to keep our convention in place then I have the rotation about Z axis as I require doing the rotation by an angle delta and then I just do the inverse of these. Therefore the inverse of R_y minus beta is R_y minus beta, inverse of Rx alpha is Rx minus alpha and the inverse of T minus P_0 is $T(P_0)$. So this gives me the entire chain of transformations which I need. And after multiplying these matrices I get one single matrix. So I can do rotation about any arbitrary axis which is very relevant and useful because you would like to have rotations about a certain axis in space and not necessarily the coordinate axis.

(Refer Slide Time: 38:34)

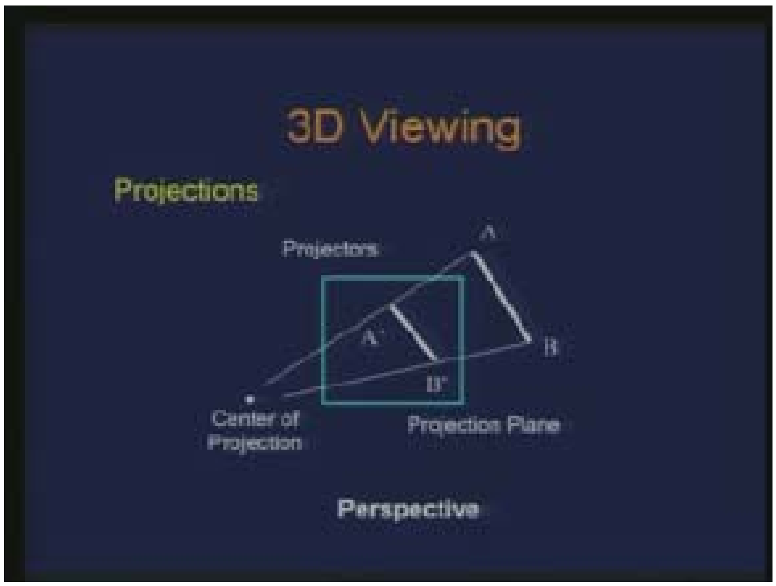


Now again we look at the general transformations in 3D basically captured by a 4 into 4 general transformation matrix where I can see certain elements of different variety of transformations we obtained.

For instance, this part gives me the translation, this part gives me the linear transformation where the diagonal gives me the scaling and off diagonal gives me the sharing and all these together may also give me rotation. This particular element gives me the global scaling of the object or a scene and these values in the last column basically are the projective terms. So this part of the matrix where we have the linear transformation and the translation combined together gives us a fine transformation.

After having seen these transformations particularly in the context of geometrical alterations geometrical changes which happened to the object we also have the set of transformations which are relevant in terms of viewing. These transformations what we have seen are basically used when I have to change the shape of the object or move the object somewhere. But I also have transformations in a very similar framework where I can apply the transformations for performing the viewing operation. Now let us try to see what we mean by viewing in general. I am basically trying to tell you what we mean by a 3 dimensional image. And then we will see the various viewing transformation in the context of 3D viewing. The 2D viewing is simple, 2D viewing is saying that I have a 2 dimensional scene and I want to view that scene in a given display. So it basically requires a window to window map, one rectangular space to another rectangular space nothing more than that whereas then I have a 3 dimensional viewing it is more involvement.

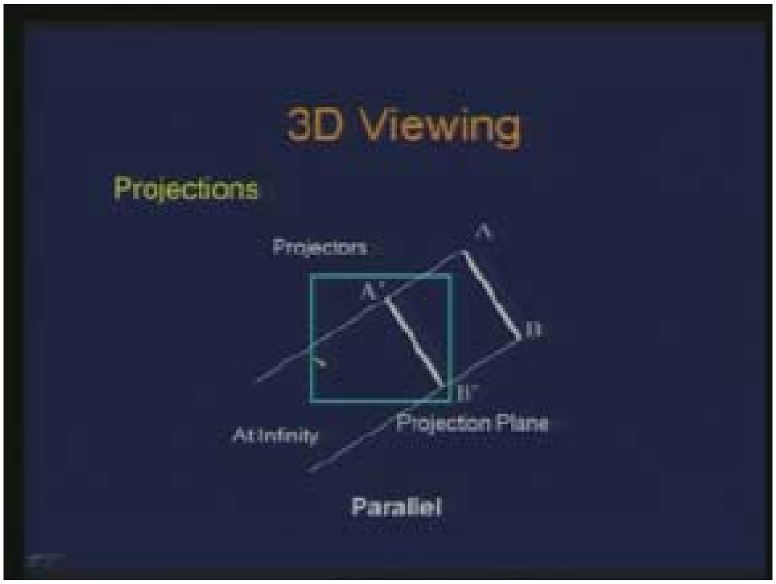
(Refer Slide Time: 42:08)



3D viewing is something what we also perform. We have the idea of what a 3 dimensional scene is and there is a 2 dimensional image which gets formed in our visual system so the analogy is similar. Now let us say I have a line AB it could just not be a line but any object or any part of the scene which is being viewed from some point here and I have the projection plane where I would like to see this line as an image on to this plane. So this is projection plane or a viewing plane or an image plane. So, when we are looking at from this center of projection and the center of projection is nothing but where if I see the rays coming from the scene or the object and in this case the line they tend to meet at some point which is the center of projection and for practical reasons we use this center of projection as the viewer or the camera.

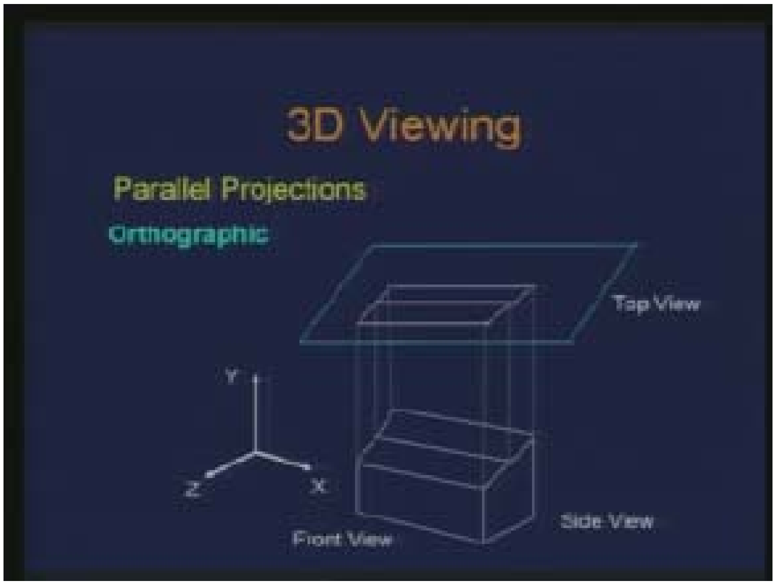
Therefore now what we have is these projectors coming from the 3D scene passing through the center of projection. And as a result I get an image here A prime B prime. Therefore these projectors which we have in this scenario are the projectors which meet at a point and they are not parallel to each other. So the type of projection we get is the perspective transformation or a perspective projection. So what we observe here is that the projectors are not parallel to each other. Here the point to be noted is that in the projectors the rays coming from the 3D scene are not parallel. Now I change this to a scenario where I have these projectors parallel.

(Refer Slide Time: 45:19)



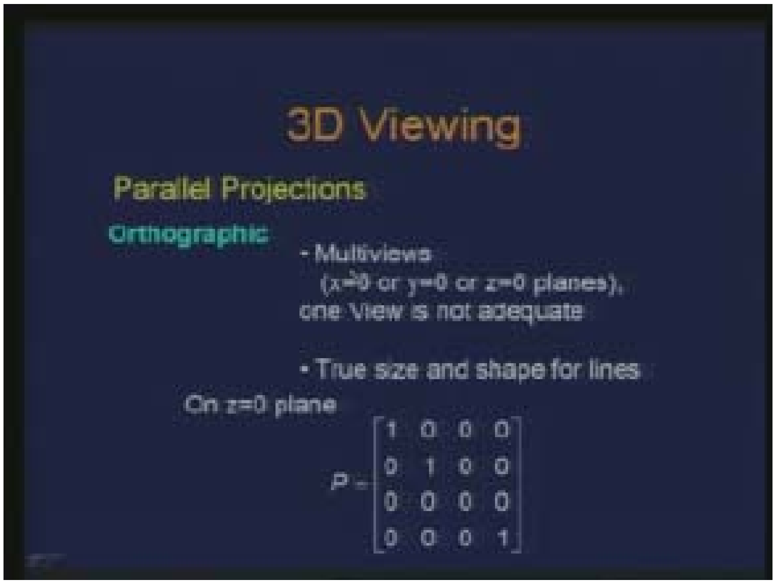
These projectors are parallel now. Then the resulting projection what I get on to this projection plane is parallel projection. And if I now see as to where these projectors meet they will actually meet at infinity. So the center of projection has now gone to infinity.

(Refer Slide Time: 46:16)



One of the very commonly used projection is the orthographic projection which is actually the parallel projection. It is a more restricted parallel projection. So here what we have is, if I have an object like this then I consider my projection plane as a plane perpendicular to one of the coordinate axis or parallel to one of the principle planes, these are all also called as principle planes which are formed by the two coordinate axes. So I consider a projection plane perpendicular to y axis then I get a projection which we referred to as the top view basically viewing from the top and similarly we have the side view and the front. These are nothing but parallel projections. You observe that these projectors are parallel.

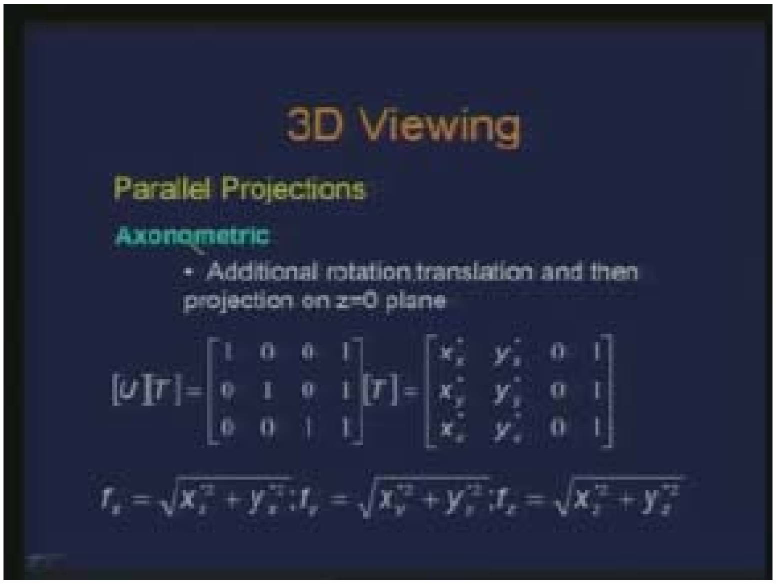
(Refer Slide Time: 47:40)



So while viewing 3D using orthographic projection we note that it may require multiple views to be able to see the 3 dimensionality of the object. So I require planes like X is equal to 0 or Y is equal to 0 or Z is equal to 0 which generates number of views where I can view the 3 dimensional objects. And we also know that the size and shape does not change using orthographic projection so the two lengths are preserved. Now if I have to perform this orthographic projection one thing we want is that we want these projections also like transformation matrix so that we can use these as transformations just the way we had looked at the geometrical transformations by multiplying by some matrix.

Therefore now if I have to perform this orthographic projection and let us say I consider one of the examples where I take the orthographic projection on Z is equal to 0 plane then what it means in terms of the matrix which I would require is this where I have the third column where all entries are to be 0. Similarly, if I have to take an orthographic projection for y is equal to 0 plane then I will have this second column where all entries are 0 and so on similarly for x is equal to 0. I can construct a matrix corresponding to the orthographic projection on to different planes. Now the problem which we observe here is that either it requires multiple views or we are not able to see the 3D aspect of an object. Now let us say I am interested in parallel projection but projections which will give me some more idea about 3 dimensional aspects of the object.

(Refer Slide Time: 50:47)

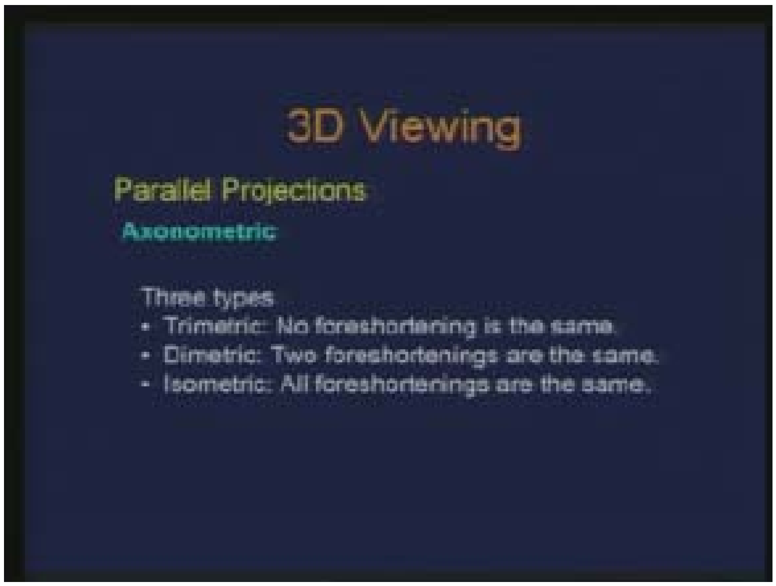


In fact there is a class of projection known as axonometric where what is done is that you have an additional rotation or translation or both and then you perform the projection of Z is equal to 0. So, by combining some transformations like rotation and translation before you perform the standard orthographic projection it can actually render the 3 dimensional aspect of an object. Remember that a projection actually can lead to certain changes in the ratios of the lengths. In the case of orthographic projection everything was preserved. here what may happen is due to this transformation they could be distortions or what we are call it as the foreshortening in each of the dimensions. Therefore in order to see that I construct this matrix T which is a combination of rotation or translation and a projection on Z is equal to 0.

This T is actually a combination of these transformations now what I am saying is that the transformation actually can give me a certain foreshortening. Now, in order to observe this foreshortening or in order to compute this foreshortening what I have is a unit vector in X, a unit vector in Y and a unit vector in Z. So 1 0 0 is the unit vector in X, I am saying that it starts from origin so I can have this in the homogenous coordinates 1 0 0 1 and similarly this is my unit vector in Y and this is my unit vector in Z. So this is what I mean here by saying U and then I apply this T which I had constructed. Now the result would give me some entries where the last column is going to be 0 because I am taking projection on Z is equal to 0 plane. So that particular column is going to be 0 and there will be some terms here.

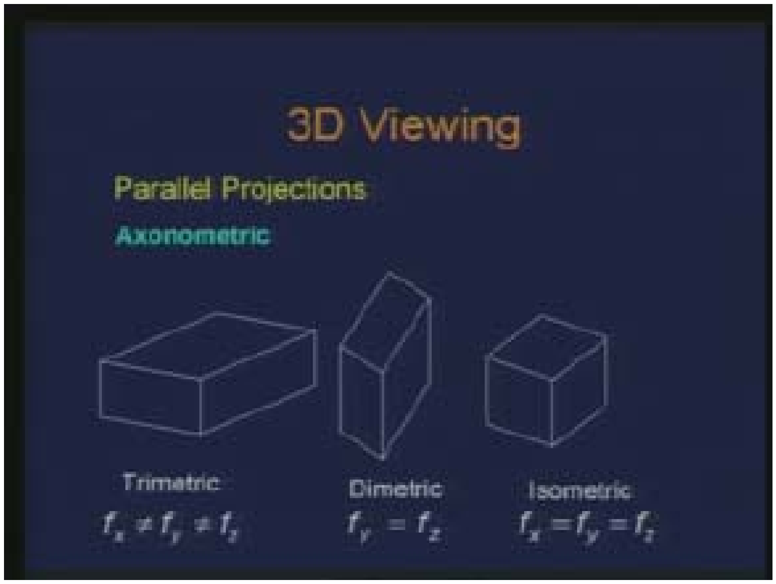
Therefore you can also look at as an individual transformation, I have put this in the form which looks like a matrix but what we are saying is that take this particular unit vector, perform this transformation T on it and you will get this. Similarly we take this unit vector you will get this. So these are the corresponding transformed unit vectors with these entries which in turn give me the idea about the foreshortening in each of the directions. What do I mean by foreshortening is basically what is the distortion which has happened in each dimension or in each direction. So, if I have the unit vector 1 0 0 transform to X_x Y_x so suffix x is basically referring to the unit vector in x direction then the foreshortening or the scaling or the distortion which has happened is given just by this which I call it as f_x so it is foreshortening in X. Similarly this is what I get as the change in unit vector in Y which I can compute as foreshortening in Y and similarly foreshortening in Z. Now, once I see that the addition of rotation translation are basically giving me different foreshortenings then I can characterize these foreshortenings to be able to define different types of axonometric transformations.

(Refer Slide Time: 56:39)



What I mean here is that basically I can define three types of axonometric projections. In the trimetric transformation no foreshortening is the same. That means f_x and f_y and f_z are different. Similarly, dimetric transformation is which two foreshortenings are the same and the isometric transformation is where all foreshortenings are the same.

(Refer Slide Time: 57:25)



This is isometric where I have f_x is equal to f_y is equal to f_z. So, all the foreshortenings are the same. In dimetric I have two of the foreshortenings as the same. So I had basically taken a unit queue and I transformed them using these projections these transformations. So here basically I have f_y is equal to f_z and here I have all of them different which I call as trimetric. So what we have is basically combination of rotation translation added to the orthographic projection where we can have the 3D viewing containing the 3 dimensional aspects. And depending on the type of the foreshortening we obtain we have isometric, dimetric or trimetric transformations.