Computer Graphics
Prof. Sukhendu Das
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture – 8
Three – Dimensional Graphics

Welcome back all of you to the lectures in Computer Graphics. Today we start the discussion on three dimensional graphics or three dimensional computer graphics. If you remember, in the last few lectures, we have been discussing various types of transformations in two dimensional graphics. Now we will see how to represent and perform those operations in 3D and that is the basis of three dimensional graphics.

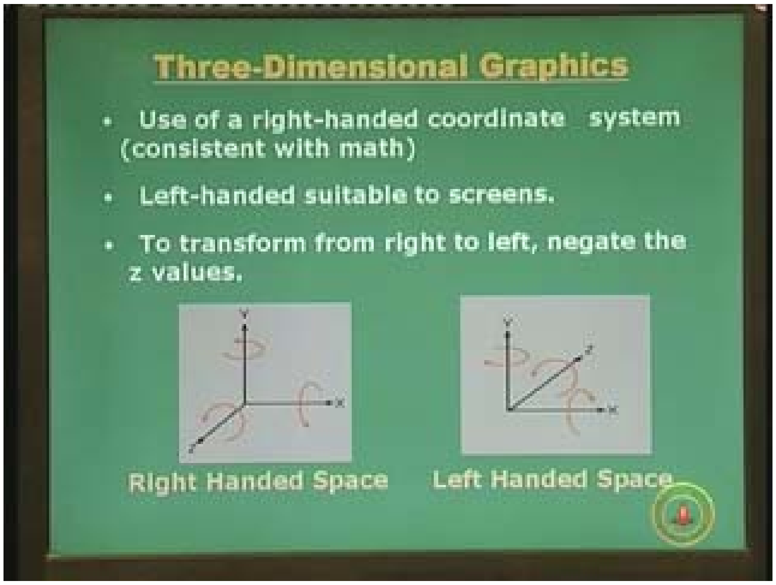
The most important difference between two dimensional graphics and three dimensional graphics is that incase of two dimensional graphics we did not have any concept of depth variations of the object with respect to the viewer. That means the entire object which we were viewing did not have any depth and it basically was on the two dimensional planar surface and we did all operations in x y plane or x and y domain.

Now we have the z component on the coordinate system, that means the object is a three dimensional object. And basically what was a square has become a cube, what was a circle has become a sphere, if it was an ellipse it has become an ellipsoid in 3D and not only that, the object has been turned from 2D to 3D but we have to worry about projection geometry for the first time in three dimensional graphics which had no scope in 2D. That means when you take an object and bring it closer to you or closer to a camera which is viewing it, the object size become larger and when you take it away it appears to be smaller due to what is called foreshortening effect.

That means larger objects may appear smaller if they are far away from you and small objects may appear larger if you bring them closer to you. A typical example could be, if you take a ball or a tennis ball or a football and bring it quite close to you in front of your face you will see a very large volume whereas we probably keep looking at the moon which is of a very very large size it typically appears very small because it is very far away from the viewer. So that is the effect of depth assuming which comes in projection geometry or what is called as perspective projection geometry.

We also have oblique projections. We also have orthographic and different types of projection geometry which we will also study. But first we will analyze the three dimensional coordinate system and see whatever transformation we have done in 2D and what happens if you just push them to three dimensional coordinate system and apply the similar transformations. So we look on the screen we first look into three dimensional graphics and we talk of a coordinate system. We usually use a right handed coordinate system as you can see in the Maths. It is called right handed coordinate system because you can open the three fingers of your right hand the thumb and the index finger and the middle finger and in orthogonal directions and each will represent x, y and z respectively.

(Refer Slide Time: 00:04:20)



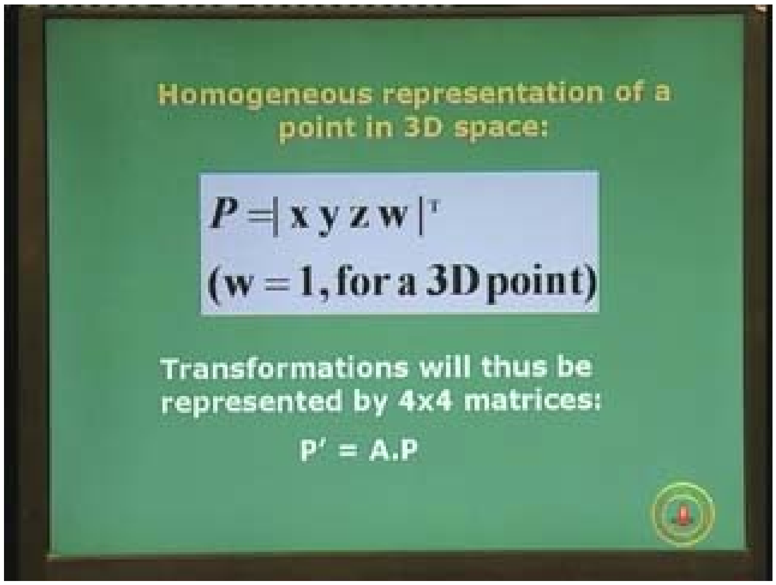
And as you see on the screen similarly you can do this with your three fingers and that is how you get a right handed coordinate system. You can also have a left handed coordinate system when you open the three corresponding fingers on your left hand, you will have a left handed coordinate space x, y and z. In 2D you only had x and y but in 3D you can have x, y and z. So you have translation along x y or z and also you can have rotations about this axis. We will see those transformation matrices later on.

The left handed system is suitable for the screen geometry where the x axis moves form left to right increasing the x values and the y axis moves from bottom to top and the z axis goes from the viewer into the screen. Whereas in the right hand space z coordinate is pointing towards you if you take the x and y coordinates system to be the screen. To transform from left to right you just negate the z values. So basically the z axis changes direction from right to left coordinate system. So assume you can visualize the two different coordinate systems. You can take any one of them but you should be consistent with the mathematics when you are using the transformations. So possibly the right handed coordinate system is very natural. If you feel that is very natural to you then you can use that. And if you remember a point in 2D was represented by two coordinates x and y. In 3D you have x y and z. I hope you have no doubt why we need homogeneous Cartesian coordinates system or a homogeneous representation of the point.

The basic reason was, you remember that we had to represent translation also as matrix multiplication like rotation, reflection, etc and so you take a homogeneous coordinate system. In 2D we had three parameter and for 3D we have four parameters where w is the homogeneous parameter which is typically taken for as one. Otherwise you divide the first three by w you basically get back the actual coordinate system from homogeneous

Cartesian coordinate system representation. And P is a vertical vector basically x y z w transpose so it is a vertical or a column vector. And when you take it as a column vector typically you premultiply the matrix P by a transformation matrix A.

(Refer Slide Time: 00:06:21)

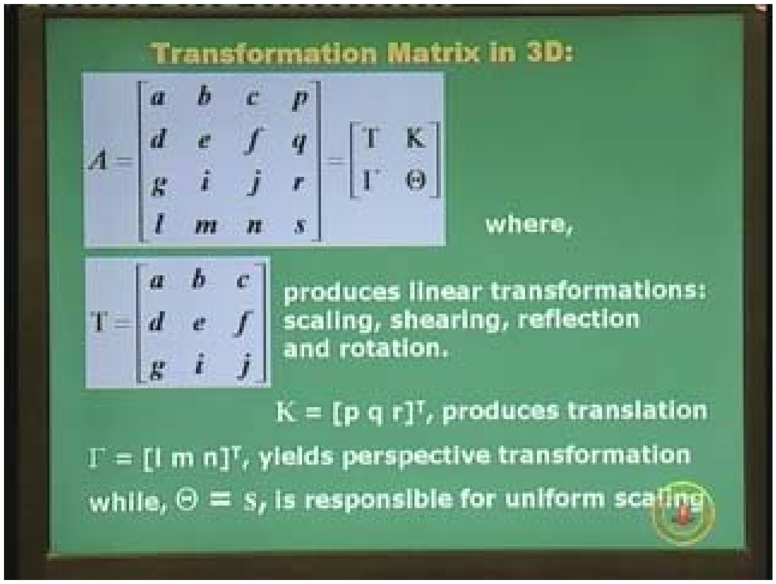


We talked about this in 2D and similar concepts in 3D as well and you get a new coordinates P prime. Of course if you take the P to be a row vector that means P is equal to x y z w without a transpose. If it is a row vector then what will happen is P prime becomes P multiplied by A or P gets post multiplied by the transformation matrix A. Now let us look at the transformation matrix A as to how it looks like.

The generalized transformation matrix in 3D will now basically be a 4 into 4 matrix as given here. So it will have 16 elements or 16 parameters in it. And you remember, in two dimensional coordinate system and two dimensional transformations when we were dealing with homogeneous Cartesian coordinates system in 2D with three parameters where a 3 into 3 or 9 elements. Now since we have four parameters including the homogeneous parameter or w as you can call it you typically need to have four by 4 or 16 elements. So if you look at the screen now, we have the A capital A matrix with 4 into 4 elements a b c p d e f q and so on. And what I do is I represent this matrix A by what I will call as partisan matrices.

Partisan matrices T K tau capital tau and capital theta where the capital T is a sub matrix or a partisan matrix of A or basically I can say it is a sub matrix of A consisting of the top left 3 into 3 elements or 9 elements form the T. And these elements of T are responsible to produce all the transformations, all linear transformations including scaling, shear, reflection and rotation. We had a 2 into 2, in 2D we have 3 into 3 which produces all your transformations and these are just the parameters.

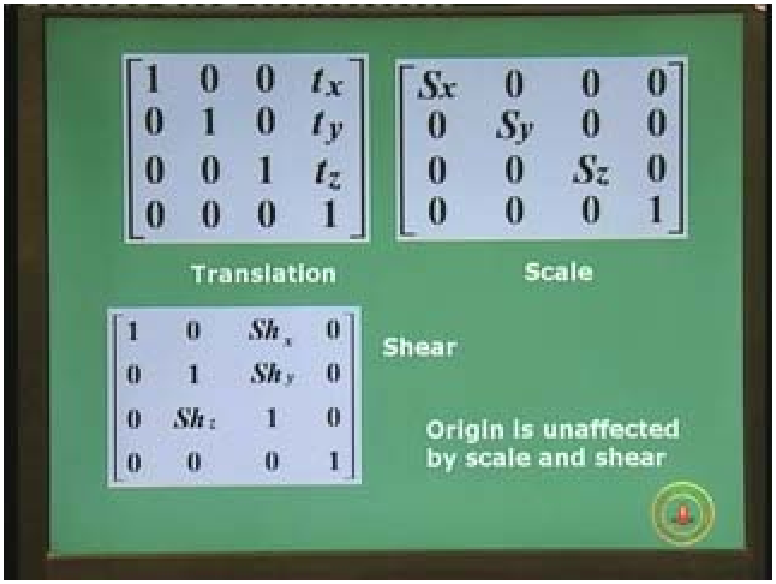
(Refer Slide Time: 00:09:58)



We will see the expressions of these T as we go along for different types of transformations. They are almost similar with an extension of one more parameter. We will basically talk more about scaling, reflection and shear. I mean now that depending upon the diagonal elements and off-diagonal elements as when it will cause shear or when it will cause reflection, scaling and so on. Now T is responsible for all this linear transformations. What about the remaining T's of matrices. We have K, we have tau and we also have the capital theta.

The K which is again a vertical or a column matrix, p q r is responsible for the translation component in 3D, capital tau or gamma as we can call. It is now a row vector l m n, it needs perspective transformation. Although you can visualize this to be a column vector but typically it is the last row of a, the first elements of the last row they are responsible for yielding the perspective transformation. So they are responsible for the parameters of perspective geometric which we will see as we go along. And the last parameter s is responsible for uniform scaling, this also show if you recollect your thoughts and concepts and equations of two dimensional transformation matrices the bottom right diagonal is always responsible for uniform scaling in 2D. In this case it is responsible for uniform scaling in 3D. And again, of course, I should repeat that you can use the parameters of the T matrix that is the upper left triangular 3 into 3 sub matrix T and you can also play with the diagonal parameters a e and j. The a e and j of the T to also use non-uniform scaling or uniform scaling also as required. But if you just have one uniform scaling you can use the parameter s without bothering about the other parameter. Now we know that upper left components are responsible for scaling, shear, reflection. The p q are right vertical parameters of the last column or responsible for perspective transformation and the last row l m and n responsible for perspective transformation, k I repeat p q r is responsible for translation.

(Refer Slide Time: 00:13:09)



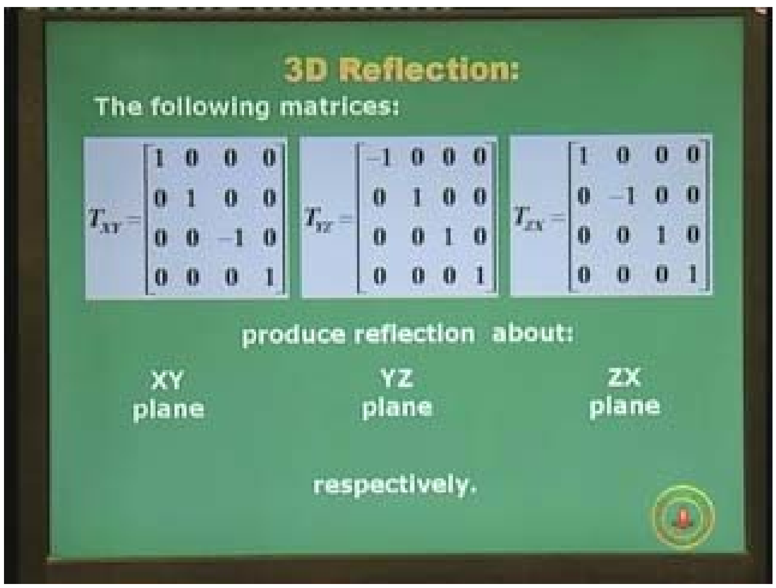
Now we will look into some special forms of these transformation matrix a and this is a simple example of translation where I change the p q r in the previous matrix to corresponding parameters tx ty tz respectively. In 2D translation of 2D transformation category you had only tx and ty that is translation along x and y respectively. Here you should have the translation parameters along x y and also z. So that is why you have tx ty tz as the three parameters for translation. You also can talk about scale, uniform scaling, uniform zooming, compression or enlargement as we talked about in 3D and in 2D, we can do the similar thing in 3D and you can play around with Sx Sy and Sz to produce non-uniform scaling along x y and z respectively. That is how you control the scaling of 3D objects with the help of three different scaling parameters.

Yes, you can use shear and as you see here the off-diagonal elements responsible for shear you can do the following by shearing with respect to x y and z. You can try these corresponding parameters in the off-diagonals sub matrix T to produce shearing effect along x y and z respectively. So please write the matrix form in terms of P prime equals A primes P when you shear along x basically what you are doing is you are creating the corresponding change in the corresponding coordinates.

Origin of course is always unaffected by scale and shears. So if you have the origin of the 3D and if you give any value Sx Sy Sz or the shear values the origin does not change. Origin of course can be translated but it cannot be scaled or sheared. So these are simple examples of translation scale and shear sub matrices. And we go into two major types of transformations; linear transformations and the upward sub matrix T is responsible for the linear transformation. So we will look at reflection matrix and the following matrices are responsible for reflection. Now you remember in 2D we talked about reflection about a line.

And here in 3D we actually talk about reflection about a plane. This is true with respect to the reflection which we actually see over a mirror and we put a mirror in front of you actually you can see reflection of yourself on the mirror or of some other object and the reflected object actually appears behind the mirror. So you are looking about a plane, you are looking at a reflection across or about a plane and so that is what you are looking at. So in 3D we talked about reflection about a particular plane and these are special reflection matrices which talk about reflection about the x y plane. So T_{xy} as given by the expression on the left hand side matrix will produce reflection about the XY plane, Tyz will cause a reflection about the YZ plane and T_{zx} will cause a reflection about the ZX plane.

(Refer Slide Time: 00:14:48)



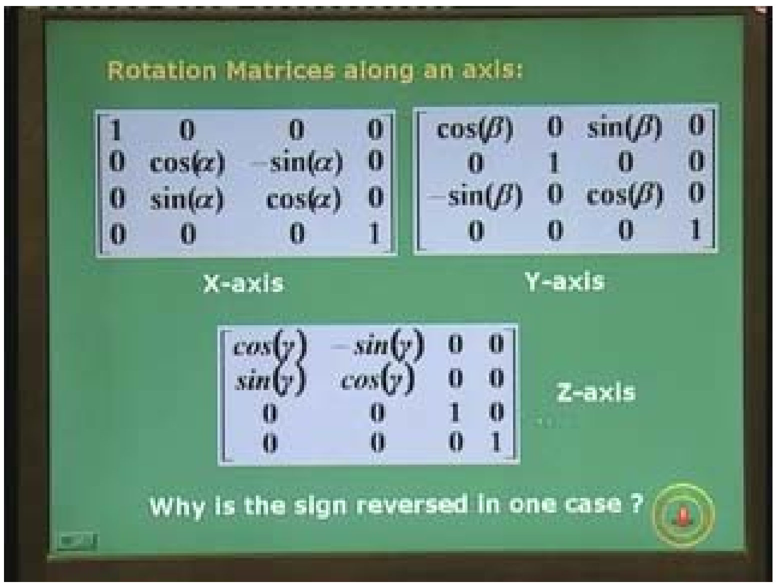
If you look at the nature of the matrix in each of these cases it is actually an identity matrix close to an identity matrix not equal to an identity matrix but it is except one of the unit values in the diagonal elements is made negative from plus 1 to minus 1 and all others are plus 1. So, depending upon which plane you want to get reflected, if you want to reflect about XY plane, you want to indicate the values of z so that is why you have in the third row, third column intersection the one change by polarity, the value of 1 changes to minus 1 or you put a minus 1 there.

If you want to reflect about the YZ plane basically the values of x coordinates of the points of the objects become negative and that is why you have the intersection point of the first row first column put as minus 1. And if you want to reflect about ZX plane you basically need to negate the values of y coordinates. If you want to negate the values of the y coordinates you have to change the second row second column intersection value which will change from plus 1 to minus 1. After the multiplication with P the y coordinates will change and become negative. So you can visualize this that the corresponding matrices will cause the reflection about XY YZ and ZX plane respectively.

These are reflections about specific coordinate planes as we had reflection about a line in 2D and we will also look into reflection about a general plane or arbitrary plane in 3D.

But before that we look into another form which is the most important part among, I should say as far as mathematics and concepts are concerned is about rotation about an axis. So rotation as like reflection also, we can form specialized rotation matrices to rotate about the orthogonal axis that is x y and z respectively. We will first look at the specialized form of those matrices which will cause rotation about x y and z. So let us look into the screen.

(Refer Slide Time: 00:22:07)



As you see the first or rotation matrix example is given, this will cause rotation about the x axis. As you can see this form is very very close to the rotation matrix in 2D. The only difference or the main difference is that it was a 3 into 3 matrix in homogeneous system in 2D. It is basically a 4 into 4 matrix in 3D and then in homogeneous you have four parameters. And in 3 into 3, if you remember there were of course cosine alpha minus sine alpha. Alpha is the amount of rotation about a particular point or the origin in 2D. Now we are rotating about the x axis. Hence, when you rotate about an x axis, you have to visualize yourself that we discussed about a right hand coordinate system. Let us say this is your x axis which is the middle finger index and y respectively, so let us say y and z.

If you look into this, visualize yourself with a scenario in your right hand coordinate system were you have the y and z as your thumb and the index finger and the middle finger represent your x axis. So when you rotate an object about an x axis. If you look if you try to visualize yourself that you are rotating about the x axis. This is your x axis; let us say y is pointing towards you and the z is pointing up. Ideally you would like to have a scenario where the z is pointing towards you. The z could be pointing towards you, this could be x, this could be y, there is no problem y is vertically up. Let us say this is x axis, if this is the x axis of your coordinate system and you have to rotate an object about the x axis by an amount theta, whatever the amount theta may be plus or minus that means clockwise or anticlockwise, you can see I will rotate the coordinate system now.

What will happen when you rotate around the x axis about a particular axis? The x axis values remains unaltered for the object but the y and z coordinates change, y and z coordinates of the object, take a particular point rotate about the x axis the x axis coordinate remain same but the y and z, this is let us say this is my y axis and this is the z axis, so if you rotate about the x axis which is around now pointing towards you the y and z coordinates change. That is why if you look into the matrix form in the screen now the second row and second column are the places were the cosine alpha and the sine alpha parameters are put basically which will cause the change in the coordinates of the y and z value depending on the amount of alpha.

Of course if alpha equal to 0 you can very well see that it becomes an identity matrix because it will not cause any rotation. So you have seen x axis. Let us look into y axis. You can also visualize rotation about y axis in a similar manner that whatever axis you are rotating about the coordinate of that axis will never, never change.

You are basically doing a two dimensional rotation in some sense. That means when you are rotating about the x axis you are doing around y and z like in 2D, x and y you had done and that x and y has become y z when you are rotating about the x axis. Now, when you are rotating about the y axis the x and z coordinates are the one which will change. And hence the parameters are put around the first and third columns and rows of the corresponding matrix because those are the coordinate values x and z respectively which will change when you rotate about the y axis of the three dimensional coordinate system.

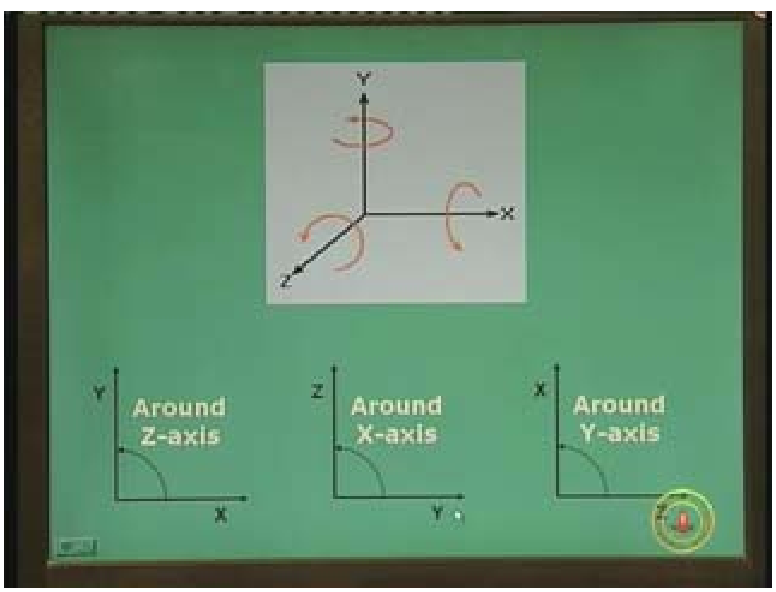
So now you can visualize that when you rotate about the z axis you actually get the form which you had seen in 2D because what will happen now is when you are rotating about the z axis the z coordinates of the point will not change and what will basically change is the x and y coordinates. This is exactly like 2D rotation where the z axis was coming out in a plane in 2D and you have the x and y on a plane and you are making a rotation about the origin with the z axis pointing out of the plane towards you or out of the screen towards you and you are moving form x to y or y to x and those were the coordinates which were changing.

So basically rotating around z axis is almost motionally coherent to the rotation in 2D which you perform because the z coordinates value will not change now but the x and y coordinates will change. So combining these three, I think it should be easy for you to visualize that I repeat again rotational about x axis x coordinates will not change but y and z coordinates will change. Rotate about y axis y coordinates will not change x and z coordinate values will change. And lastly rotation about z axis will cause change in the values of the x and y coordinates but not the z coordinate. This is the concept of rotation around any one of the orthogonal coordinate axis.

An important property where we have to spend time here, I am not sure whether you have noticed is with respect to these three matrices which have been basically borrowed form the concepts of 2D rotation, there is one sign of alpha or beta or gamma where the sign is changed and that we know why because this concept came from 2D. But you see here, the position of the sign is different in one of the case. That means the sign will reverse in one of the cases, but why? Let me try to give you what I mean here. If you compare the position of the negative sign of the sign of the angle of rotation it is at the same position with respect to I am talking of the relative position of the cos alpha sine alpha terms. Just consider the cos and sine terms and do not worry about the zeros and ones in the corresponding matrices.

If you see the matrix form for rotation about x axis and if you see the corresponding one with respect to z axis they are almost similar except the position depending upon the axis. But the negative sign is the same that is on the top right sine of any angle. But if you see the rotation about the y axis the sign is here, the sign is here on the bottom left instead of being on the top right. You see with respect to the x axis it is top right with respect to z axis also is top right but with respect to y axis instead of being on the top right here it is on the bottom left. Now there is a reason, it is not a mistake, this is a reason why this is so. We will spend a little bit of time for you to know why this is so because otherwise if you write these equations and implement using a program in C you will make a mistake by putting the sign at same position in all the three cases and that is not right, so let us see why this is so. We come back to three dimensional right handed coordinate system as you can see here x y z and those arrows mark the rotation.

(Refer Slide Time: 00:28:23)



Now let us take the rotation about the z axis. We rotate about the z axis, the rotation is from x to y. This is given in the three dimensional figure here or also on the left hand side bottom figure where you have to visualize that the z axis is coming out of the screen towards you. So when you rotate around the z axis, if you carefully see the rotation sign now, the curved arrow is put in the three dimensional screen. You can visualize this in the following manner. I will ask you to do the following. In the right hand coordinate system what I will ask you to do is you grab this axis by the right hand.

Let us say this was the axis which was pointing say it could be x y or z or whatever. This is the axis which is going and you grab the axis yourself using the right hand, the direction of the thumb points to the positive direction of the axis, it could be x y or z. Suppose if this is the axis pointing upwards, if you grab the axis the direction of the thumb will point towards the positive direction of the corresponding axis and the direction of the rest of the fingers will actually point in the direction of rotation. That means if you take any one of the axis x y or z of your right hand system, grab any one of this axis with your right hand the thumb points towards the direction of the positive axis away from the origin and the rest of the fingers point in the direction of rotation. This is one conversion which is used for right hand system.

The other way of visualizing is; if you look towards the origin from the positive direction of any axis the direction of rotation will be counter clockwise for you. That means if you are looking towards me let us say the origin is towards me, you are looking towards me from the positive z axis or positive x axis whatever the case may be and the direction of rotation will actually be counter clockwise for you. This also you can see from the figure that there are two ways of interpreting the positive direction of rotation; either you grab the axis and then the fingers point to the direction of the positive part of the axis, the rest of the four fingers point to the direction of rotation or you look down towards the origin.

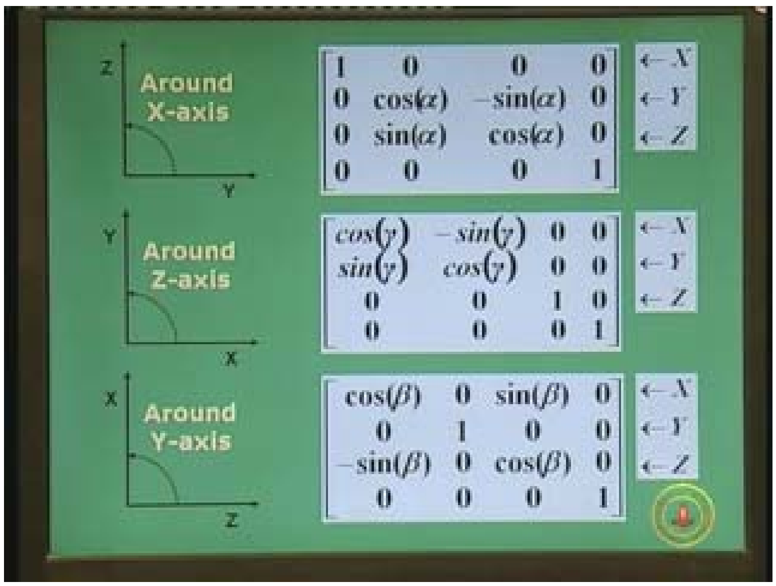
When you look down you will see that it will actually be in a counter clockwise direction. You have to practice yourself if you have not done so earlier in any course of three dimensional geometry or courses related to computer graphics. And come back to the screen you will see this picture once again, the direction of the arrows are marked in the direction of the rotation. When you grab any axis x y z the thumb will be pointing outwards, the rest of the fingers will point in the direction of the rotation. Or you are looking towards the origin from the outside you will see that the rotation actually is counter clockwise.

Look into z axis, look from the z axis positive towards the origin you see counter clockwise. Look from x towards the origin you see counter clockwise. Now if you put that into perspective and come back to the figures at the bottom, so you are now looking from z towards the origin, on the left hand side figure if you look at the rotation around z axis this is the counter clockwise rotation marked. So you are basically rotating the object from x towards y remember then. That is the conversion of positive rotation based on which the polarity of the cosine and the sine alpha terms are basically used. Now this is the rotation about the x axis.

Rotation of the x axis will cause the counter clockwise rotation from y towards z. And also hence forth rotation about y axis will cause rotation from z towards x. This is at par with the 3D system which you see on the top. I repeat again, rotation around z axis will cause rotation from x to y. Just remember these sentences and we will see why the sine comes out. Rotation about x axis will cause rotation from y to z and rotation about y axis will cause rotation from z to x.

If you have understood the convention of positive rotation by the thumb rule sort of a thing all looking down towards the origin from the positive axis and then looking counter clockwise from positive rotation then we will now look into the transformation matrix and see where the sine is kept to ensure that this consistency is kept in the three rotation matrices for rotation around z, around x and around y respectively, here at the three rotation matrices once again.

(Refer Slide Time: 00:29:23)



As you can see on the left hand side I have put those three figures, you see the first column, rotation around x axis y to z, rotation around z axis from x to y and rotation around y axis form z to x respectively.

So, if you look at the first row, when you are rotating around the x axis this is the corresponding transformation matrix. The x coordinates is unchanged so that is why you have 1 0 0 on the first row and you want the coordinates of y and z to change in such a manner that the rotation is taking place positively counter clockwise or intuitively let us imagine from y to z. And hence for these parameters cosine of alpha minus sine alpha sine alpha cos alpha will cause rotation from y to z. Intuitively visualize yourself that this causes from y to z. If that is so, then rotation around z axis from x to y, z axis from x to y will also cause this where you have the 0 0 1 at the third row where you do not want any change of the z coordinates of the object. You just want the x and y values to change that is intuitively from the rotation which is taking place from x to y and previously it was from y to z but now it is from x to y.

Now, if you look at around the y axis the rotation which you want is actually z to x. If you want from z to x not from x to z to maintain the consistency of positive rotation, so if we have put minus sign here like as in the previous case, what would have happened is, you would have forced a positive rotation to go from x to z, that would have violated the principle of right handed coordinate system or the consistency of positive rotation or counter clockwise or whatever you have studied. If you had put the negative sign in the first row of the third matrix it would have resulted in a rotation from x to z. But you want the rotation from z to x. If you want rotation from z to x put the negative sign here. It is actually something like talking of a clockwise rotation and the negative rotation in some sense to maintain the consistency.

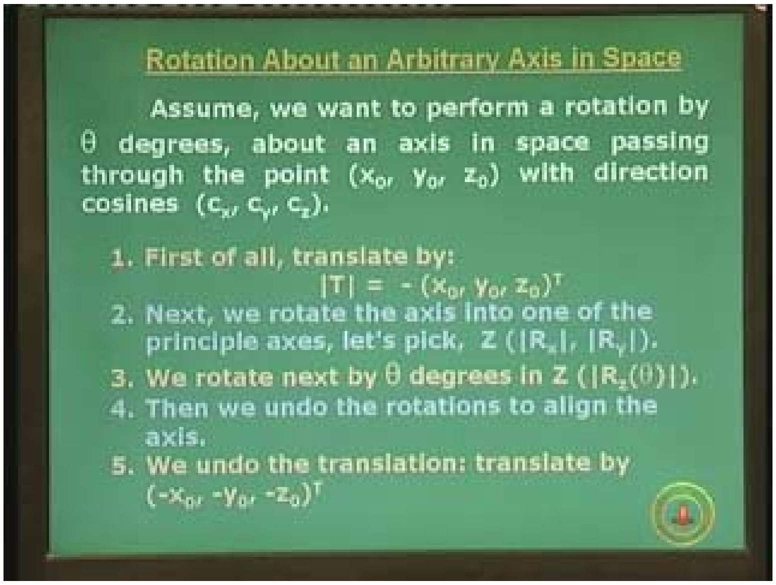
In the first case I repeat, you want y to z, second case you x to y, in third case you want z to x not x to z. Since you want z to x you have to put the sign here. I repeat the first case again, from y to z so that is why y has the negative sign here, the second row, second case around z axis x to y so that is why the first row for the x axis has a negative sign, here you want z to x that is why the third row for z has a negative sign.

I hope, if you play this around yourself with the whole thing again which we talked about last ten minutes, starting form the right hand coordinate system, talking about gripping the axis and talking about the thumb pointing upwards to the positive direction, the fingers pointing to the positive direction of the rotation or looking down to the origin with counter clockwise conversion of positive rotations. To maintain that consistency with respect to the entire axis you need to have one of the signs reversed for the rotation matrix. Any one sign reversed should have been ok but here it is done for rotation about the y axis but for which you actually want the rotation to happen from z to x instead of x to z.

I repeat again; if you had put a negative sign here like the previous two cases, in the third matrix first row if you have put a minus sine beta you would have a rotation from x to z which you do not want. That will violate the consistency or right hand coordinate system when you want to rotate from y to z, then you want form z to x and then x to y respectively. Now that consistency is very very important. So these are the three fundamental rotation matrices about x y and z respectively.

Now, if you remember in 2D transformations we also talked about rotation about an arbitrary line, rotation and reflections about arbitrary lines and planes and first of all general and very special cases and then we moved to a general rotational matrix. So in the next few minutes we will study rotation about an arbitrary axis in space.

(Refer Slide Time: 00:36:00)



So what do we have here; assume we want to perform a rotation by an unknown theta about an axis in space and now an arbitrary axis because we have seen rotations about three specific coordinate axis x y and z separately, respectively we have seen the forms.

Now we are talking about an arbitrary axis. You remember in 2D we talked about rotation about an arbitrary point in 2D. Here we have rotation about an arbitrary axis, why an arbitrary axis? Yes let us take the earth, it also rotates around the sun, it also rotates about its own axis, the moon also rotates about the earth. So when you talk of rotation you do not talk of rotation about only a point as in the case you did in 2D. In 3D you have an axis and you rotate about that axis. So that axis was very specifically the orthogonal axis x y and z respectively, the three corresponding axis in the right hand system.

Now if you have an arbitrary axis the problem could demand to rotate about an arbitrary axis and we need mathematical formulation to rotate an arbitrary object or a point about an arbitrary axis in space. That is what we are studying now. So I again repeat the problem as given in the slide, we want to perform a rotation by theta degrees about an axis in space and that axis is defined by the following parameters. That axis passes through the point x_0 y_0 and z_0 and it has its direction cosines of the axis. I will define what direction cosines are. For the time being assume that they are something like cosine of the certain angle with respect to x y and z respectively. And the direction cosines of that line are defined as cx cy and cz respectively.

So there are six parameters which define a line uniquely. You need a point about which that line or axis pass through and the direction cosines or orientation of that line. Once that line is uniquely defined, now we will try to rotate a point or an object about that particular axis. So let us come back to the screen. These are the different steps which we have for rotation about an arbitrary axis in space. It is almost conceptually similar to what we did in 2D rotation about an arbitrary point.

You remember what we did, when you wanted to have a rotation about an arbitrary point we translated the point to the origin or origin was translated to the point. We did the rotation and then translated to the point back again.

So here it is; you may have few more steps. But actually if you see the steps 1, 3 and 5 translate, rotate and then undo the translation. Actually it is the same as like in 2D. Since we have 3D here, there are two more steps, the second and the fourth steps where we need to align the axis of rotation with respect to the coordinate axis because in 2D we had the special form of rotation about the origin so that is why we brought the origin to the point. Here we have the form of the rotation matrix to rotate about an orthogonal axis, any axis x y or z.

Now we need to first translate the line back to the origin or origin to the line and then orient this axis such that it coincides with one of the axis. It could be x y or z. Once we do that we can apply our regular specialized rotation matrices and get the transformation done. So let us read the lines, the steps of the algorithm. First of all we need to translate by a certain amount which is nothing but the x_0 y_0 z_0 the coordinates of the point about which the axis in space passes through.

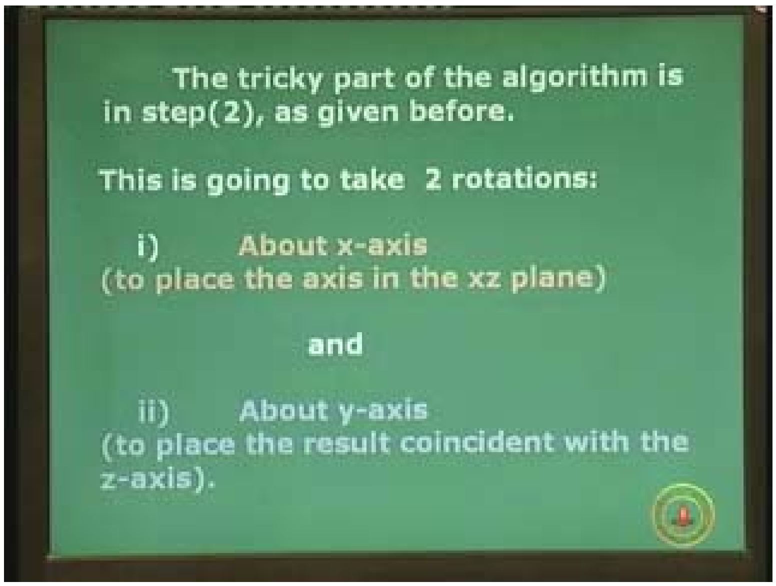
Next, we rotate the axis into one of the principle axis and we will see why we actually do that by a rotation about x and y to coincide the axis with the z axis. And once the second step makes the arbitrary axis coincide with one of the principle axis in this case it is z, we need to employ the rotation about by theta degrees about the z axis and once they are done we have to just undo the steps 2 and 1.

In the opposite sense, you remember the matrix multiplication order is very important. So step number 4 says we undo the rotations to align the axis. That means whatever we did in 2, we just do the reverse operation which can be done by inverse of the matrices which we have already seen in 2D. And last point of course and the operation done in step number 1 are basically to undo the translation. These are the five steps and typically I think we know step number 1, we know step number 3 that is we know the form of the matrices in these cases and we know what it does because it exactly does what we did for the case for 2D. We also know step number 5 because it is nothing but the inverse transpose or inverse translation of step number 1.

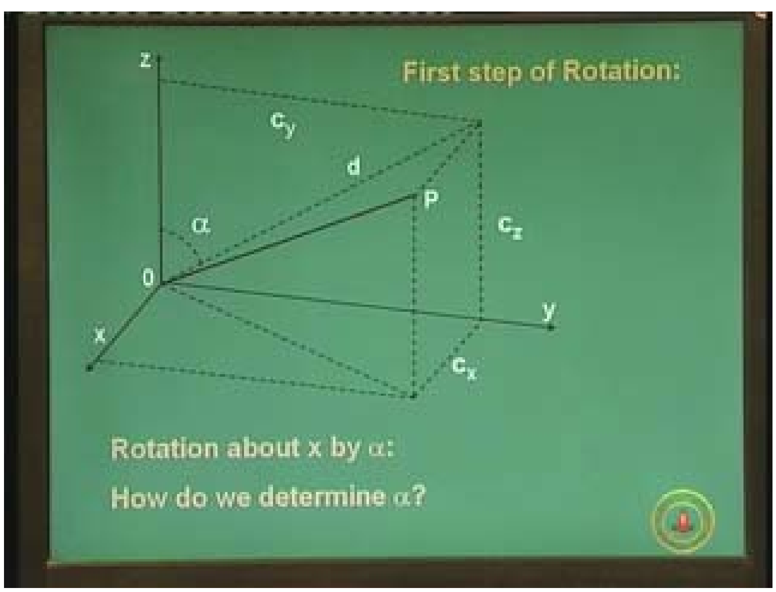
Step number 2 is important, that is the one which we have to solve because if we solve step number 2, step number 4 is just an inverse operation of step number 2. Once 2 is solved we can get 4 and steps 1, 3 and 5 are all known to us. So we concentrate only on the operations to be done on step number 2 and then we can get all the matrices together and get the final form. Remember.

So we go to the next slide where we concentrate on what to do for step number 2 which reads; we rotate the axis onto one of the principle axis and in this case we pick up z which means we need to rotate along x and y respectively. So as I said before, that the tricky part of the algorithm is basically step number 2 as given before in the previous slide and this is going to take us two rotations; one about the x axis, I will show you the figure why I say so, that you need to rotate the arbitrary axis about the x axis to place it in the xz plane first and then you rotate around the y axis to place the result coincide with the z axis. Actually you can do the other way, nobody stops you from first rotating about the y axis to coincide it along say xz plane and then you can do the reverse.

(Refer Slide Time: 00:39:26)



(Refer Slide Time: 00:42:01)



So let us look at the first step of rotation which we said that you need to rotate around the x axis to put it back into the XZ plane. If you remember the previous example, I was saying that you rotate about the x axis to place the axis in the XZ plane. Although it could be the other way round where you can first rotate about y axis and then pickup the x axis but we pickup the x axis first in this case.

So what do this mean? Let us look into the figure again, op is the line you can see in 3D diagram the first of course we have a 3D diagram although you are seeing it on the screen which is 2D but you have to visualize this is a 3D diagram. You have to visualize x z and y the three perpendicular orthogonal coordinate system axis. You can start to imagine x is pointing towards you and y and z and let us say the screen vertical and horizontal coordinates respectively and op is an obliquely oriented axis about which we actually have to rotate. So we have come to the stage where the origin is shifted to some point on the op axis. And now we have to do something such that this line or axis op coincides with oz.

I repeat; we have to do two rotations to make op coincide with oz. Well we can start thinking that it could be done by one rotation but you have to do some mathematics which is equal to performing two rotations. So op has to coincide with oz and that is done first by rotating around x axis so that op will first fall on the X_Z plane and then you rotate around the y axis such that the projection of op which has fallen onto the X_Z plane comes back to oz. So the two steps of rotations of op axis will bring that op coincides with oz. The first step is rotation of x by alpha, how do you determine the angle alpha?

Now you see, I have projected the axis op onto the Z_Y plane and C_X C_Y C_Z are nothing but the direction cosines. So assume that the ops are of unit vector in length and the corresponding coordinate projections are C_X C_Y C_Z respectively which are nothing but the direction cosines and d is the diagonal or the length of the projection of op on the Z_Y plane when you project op on the zy plane.

When you project it remember, but what we are basically interested at, we are actually not going to project, it is just a projection. That means if I rotate op by an amount alpha which will bring op onto the X_Z plane it is equivalently you can visualize that this diagonal line which is marked as a distance d on the Z_Y plane will coincide with the z axis. So when you turn op and coincide it with the Z_X plane the line marked d will coincide itself with the oz plane. This is what will happen, we are talking about positive rotation about the x axis, grip the x axis thumb pointing outwards and look towards the origin and visualize counter clockwise rotation. op will now coincide with X_Z axis and d will actually fall on oz and that is what will happen that gives you the alpha. So instead of measuring alpha directly from op with respect to z axis we can measure alpha from the diagonal line d with respect to z axis and so we can see what is alpha, so that was the step which we are talking about.

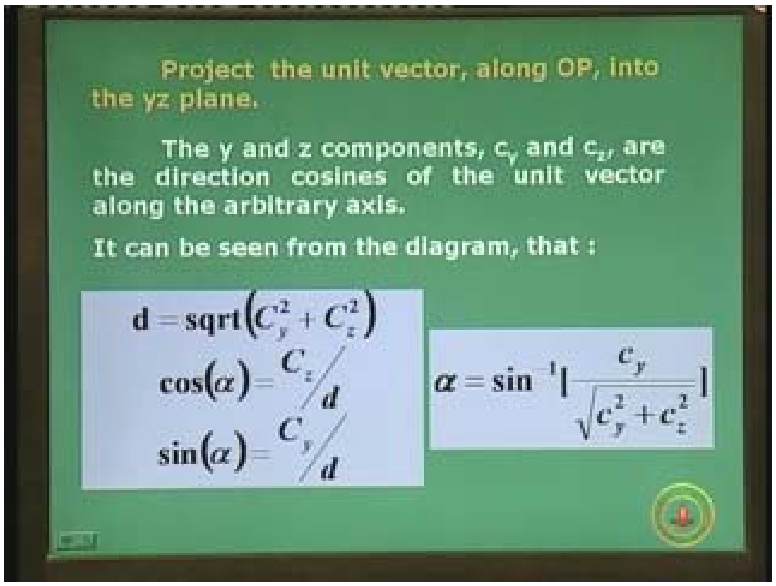
Project the unit vector along op onto the YZ plane and the Y_Z component as you see in the previous figure. I hope you have drawn it on your notes already. We will also go back to that figure, it will come again but the Y_Z components of that projection after projecting onto the Y_Z plane at C_Y and C_Z respectively which are nothing but the direction cosines of the unit vector along the arbitrary axis we also talked about that. And it can be seen from the diagram that I will give you the diagram and give you the equations.

Once again see the diagram here. So we can see d C_Z and the corresponding C_Y. I can say C_Z here C_Y and d actually form a right angle triangle. So this helps you to evaluate the value of d. And that is given by the following expression, so d is given by square root of or root over cy square plus C_Z square or you can now actually evaluate alpha using C_Z by d cosine inverse or sine inverse.

And you remember, we need both cosine and sine to evaluate alpha not only it is remadical value but the sine. If you take only cosine or sine due to the bipolar nature of the cosine and sine functions either only cosine or sine will not give you the true value of the alpha. The sine of the alpha value means that in which quadrant the alpha is? You will basically have a problem, you will need a combination of cosine and sine alpha. This you can get from pure mathematic Septuagint trigonometric knowledge that you need basically both cosine and sine alpha. They actually get the sine and also the numerical value of alpha.

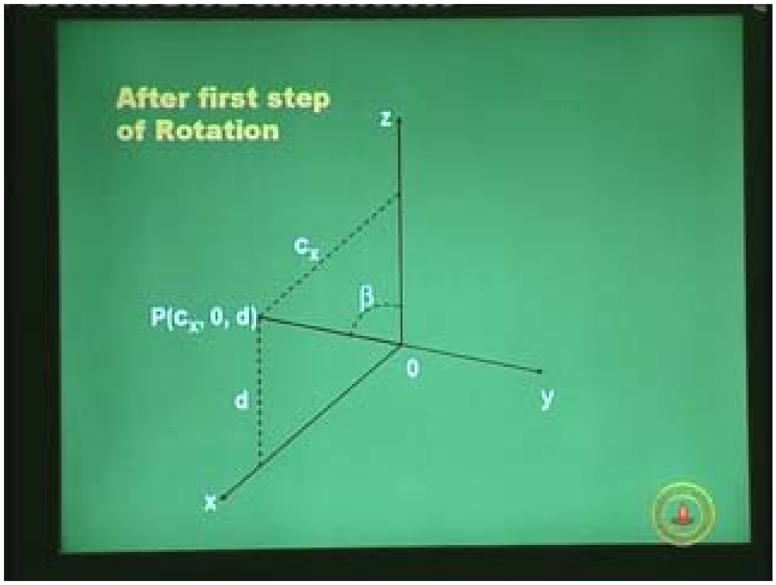
We are looking at alpha which can be obtained using sine inverse of this function. You can substitute d and that is the value of alpha which you get. Now that means when you rotate the axis op about the x axis by an amount alpha the op would be now lying on X_Y. It will be lying along the X_Y plane, we will see what it results in. This was the starting figure if you remember. So I should correct myself that when you rotate op around x axis in a counter clockwise fashion it will fall on the X_Z plane and the previous slide now helps us to know the value of alpha.

(Refer Slide Time: 00:42:01)



So the stage after the first rotation will look like this, this is the new figure. After the op has gone onto the new position which is given by P(C_x, 0, d) that means P has basically fallen on the X_Z plane. This will happen after the first rotation about the x axis. Now we just need to rotate the op axis about the y axis by an amount beta to coincide the op axis with respect to the z axis.

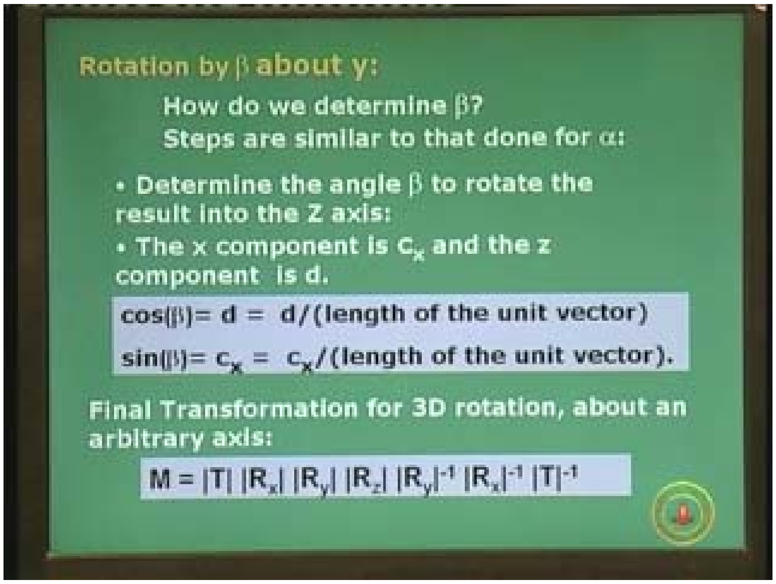
(Refer Slide Time: 00:42:01)



Remember that was our target of step number 2. Step number 2 was to take the axis op and coincide with the z axis. This was the second step of the general rotational matrix and to do that the first step of that second stage of the algorithm is to rotate about x which will give us this particular scenario. So alpha we know now we need to just calculate beta in fact it is done almost in similar manner.

Since op is a unit vector you can imagine and this length will still be the same and since this is C_X and obviously this value has to be d. The d value which was here on the Z_X plane, this is the projection of op on the Z_Y plane is now nothing but the vertical coordinates of the z coordinates of op after the projection because root over c square plus d square should be equal to be 1. So what is the second stage of rotation? Rotate beta rotate about y axis by an amount beta, and how do we determine beta? The steps are similar to what we have done for alpha.

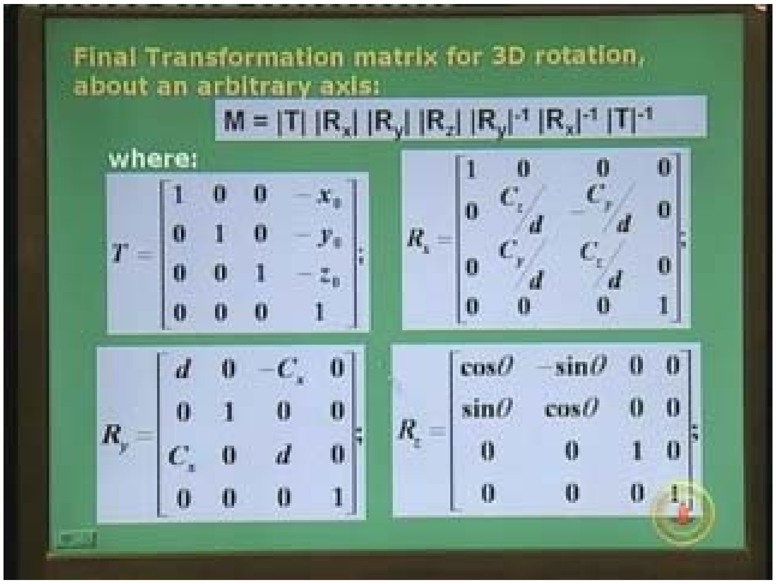
(Refer Slide Time: 00:48:16)



And first is determine the angle beta to rotate the result about the z axis and the x component is C_X and z component is d. And the cosine of beta is d divided by length of the unit vector is equal to 1, sine beta is this and so beta can be evaluated using cosine beta and sine beta and this gives us the R_x and R_y respectively for the second stage. The third stage is R_z and the last stages are the corresponding inverse matrices.

So the final transformation of 3D rotation about an arbitrary axis is first translate along T then two rotations about x and y given by the corresponding values of alpha and beta respectively. And once this R_x and R_y have been operated what basically remains is the rotation about the z axis by an amount theta and this is step number 3. And then you have to undo whatever you have done earlier in terms of R_y and R_x so apply the inverse transformations and then apply the T. So we are basically talking of post multiplication of the coordinates and this will be the order. So apply T first, if it is premultiplication the entire order will be changed. The leading matrix will be T and then R_x R_y and so on. This is probably a case which I am talking about post multiplication of the point by the corresponding transformation.

(Refer Slide Time: 00:50:27)



So I write it again, you remember the steps? First translate then corresponding two rotations, then the actual rotation of the theta and then just keep undoing the two rotations about x y and T respectively. Now I must give you the form of all those matrices here T, R_x and R_y respectively.

T of course was given to you earlier, x_0 y_0 z_0 which was nothing but the point on the axis to where we have to shift the origin. R_x and R_y were obtained from the alpha and the beta which we just talked about while trying to align arbitrary axis op with respect to z. So you had R_x and R_y respectively and these are the corresponding rotations. Please write those equations and practice it yourself.

Remember, R_x is a rotation about x so the parameters occur in the second and third row and column. Rotation about y axis will keep the second element 1 and first row and third column elements are given by rotation about beta. And of course the rotation about the z axis is the one where the actual rotation takes place about theta.

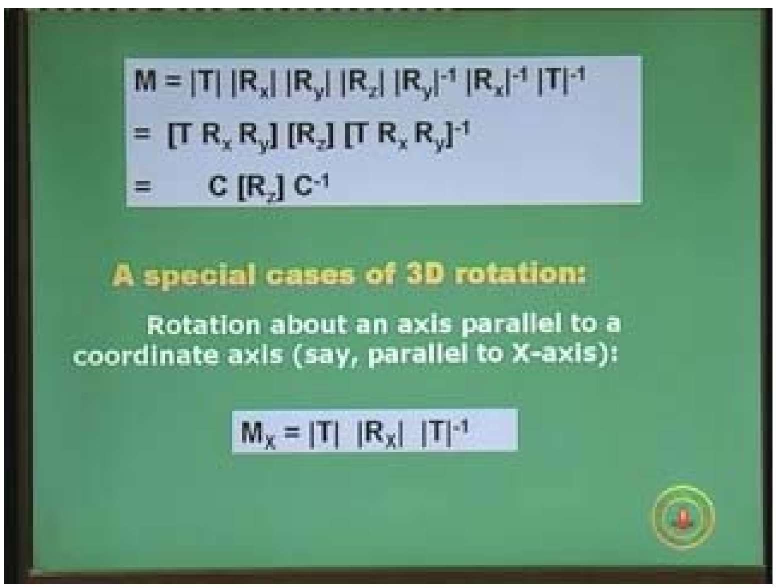
So once you get the first three or four rotations you can know how to get R_y inverse, R_x inverse and T inverse is nothing but all these are the same but just the signs will change. The sign will change here in T and in R_x and R_y since they are orthogonal matrices of rotation as in the case of 2D you get the R_x inverse and R_y inverse to be their transpose. To be their transpose of the corresponding matrices and for the case of T the sign will change.

I request you to come up with the general form of the matrix M by taking all this formulations of T, R_x, R_y into account and it will give you a good practice of analytical algebra in terms of matrix multiplication, manipulation and try to get the general form of the matrix M. I have purposely not given it here, I leave it as an exercise for you for this particular chapter or lecture that is try to get the elements of M.

You will have a 4 into 4 that is 16 elements of M and try to get all the 16 elements of M by multiplying. How about matrix multiplication? About seven matrices are to be multiplied which are all given here, four of them are given here, the other three can be obtained by just clearly making the inverse of the R_x and R_y respectively. That is how you get the generalized rotation about an arbitrary axis.

I hope the concept is clear, you basically rotate, you translate the origin to that axis or take that axis back to the origin then what you do is basically give two rotations to align the axis with respect to the z axis. It could be x or y but in this case I have taken z, rotate around z and just do the inverse on manipulations of the previous two stages to get back the final form. This is how it will look like; I am trying to make the matrix look slightly simpler. The generalized transformation matrix for rotation about an arbitrary axis C and C inverse are nothing but T, R_x, R_y multiplication respectively.

(Refer Slide Time: 00:52:38)



Special case of 3D rotation: If you try to look at, we are looking at rotation about an axis parallel to a coordinate axis say x axis if you want to rotate then you do not almost need anything. What you basically need to know is you need to translate the optical axis op or the arbitrary axis op to the origin and the axis is automatically coincide with the x axis we apply the Rx rotation rotate about y axis. A special case of 3D rotation if you want to rotate about an axis parallel to say y axis apply the same T apply R_y and apply T.

Rotate about an axis say parallel to z axis, well T again R_z T. So these are the three special cases again of 3D rotation. If you want to rotate about an axis which are parallel to any one of the three coordinate axis that is easy to visualize. On the top of the slide here you can see you have the generalized expression given and that has been the main part of the course today.

I will stop with this lecture for the one hour where we have started with three dimensional transformations, we looked at coordinate spaces first and we looked at specialized transformation matrices for translation, scaling, shear and then rotation. And we spent the rest of the time on trying to describe how to obtain a rotation about an arbitrary axis in space. Please go through those transformations once again. Draw those diagrams and write down all the equations of transformations, try to create, I left these as an exercise for you to come out with the expression of the value of M for all the these 16 elements and that will give you a good grasp of the problem which we have discussed today.

We will stop today and continue in the next lecture on the remaining side of the transformations. Thank you very much.