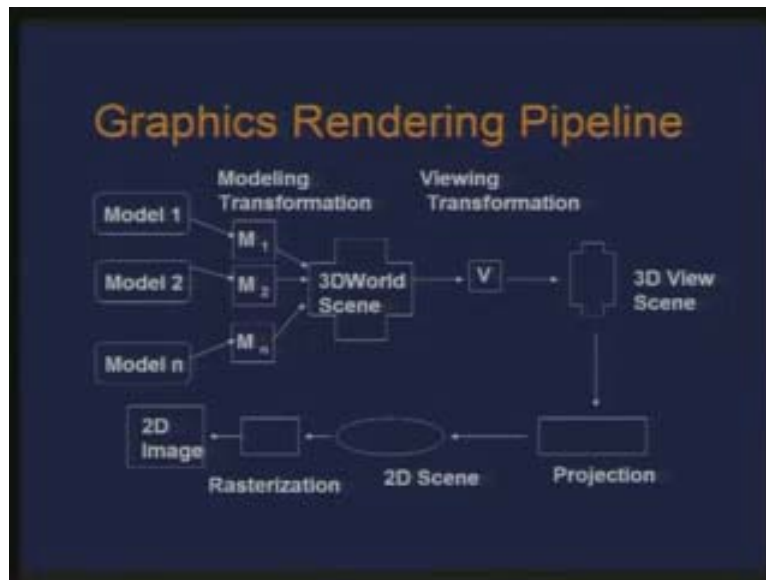


Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture - 6
Transformations

Today we are going to talk on a new topic called as transformations. So let us revisit the rendering pipeline.

(Refer Slide Time: 1:12)



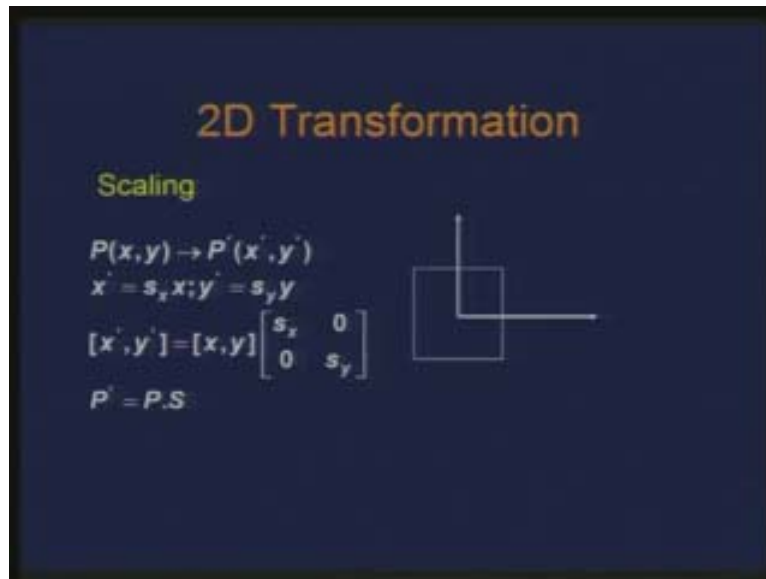
Basically when we look at the rendering pipeline we have the model generations stage where different models can be combined together using what we call as modeling transformation which are nothing but a sort of instantiation of these models to build a three dimensional world or a scene. So we need some transformation which would take these models to build this 3D world scene. And then this three dimensional world scene through a viewing transformation, a viewing transformation basically incorporates a viewer or a camera in the scene and then with respect to that viewer we generate that world whatever has been constructed.

After this viewing transformation we again get a 3D scene which is having a reference with respect to the camera. The point here to be noticed is that there is the transformation which is happening from here to here. And similarly from this 3D view scene we then project to come to a 2 dimensional scene so that again requires a transformation and then we may actually do the clipping in that 2 dimensional scene.

We can also do clipping in 3D scene. So this projection box actually consists of doing the projection that means reduction of the dimension and also clipping which gives us basically the 2D scene. And then through the process of rasterization we convert this 2

dimensional scene which is device independent to an image to be rendered on the screen of the monitor. We have already seen the rasterization process, we have also seen some component of this projection which is the clipping part and now what we are going to look at is the transformation which is happening in this entire pipeline. So we notice that there are number of transformations which can happen here. So we start with a simpler scenario where we will look at these transformations as you are perhaps converse with it the geometrical transformation. And we start these transformations in 2 dimensions.

(Refer Slide Time: 4:27)



Therefore let us see the various transformations in 2D that are useful in the context of computer graphics. For instance, let us take scaling. there is this point P having the coordinates x, y which is transformed to P prime having the coordinates X prime and y prime after a scaling transformation. So when I say there is a scaling transformation what I basically mean is that the X prime is in the form of some scaling factor S_x applied to X and y prime is some scaling factor S_y applied to Y that is what gives me the scaling.

In other words this transformation from xy to X prime y prime can also be represented in the form of a matrix where I find the diagonal terms to be the scaling factors. I can actually represent this whole transformation in this form. There is a matrix here which I represent by S then this P prime is nothing but P multiplied by matrix S. So if I have a figure like this; for the purpose of simplicity I assume that the center of this figure is actually at the origin and if I perform scaling to this figure which is some sort of contraction of that figure then I would obtain something like this.


(Refer Slide Time: 6:21)

2D Transformation

Scaling

$P(x, y) \rightarrow P'(x', y')$
 $x' = s_x x; y' = s_y y$
 $[x', y'] = [x, y] \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$
 $P' = P.S$

$s_x = s_y$: uniform scaling
 $s_x \neq s_y$: differential scaling




So, expansion and contraction are the kind of things we can obtain using scaling transformation. Now the point which is to be noted here is that if I have the two scaling factors given by S_x and s_y to be the same then I obtain uniform scaling. If they happen to be different then I have a non-uniform differential scaling. This scaling is actually happening about the origin in the sense that if I have the figure somewhere there then you would have noticed not only the contraction but also some sort of a shift if I was doing scaling about the origin. So here I have deliberately assumed that the centre of this figure is actually at the origin.

(Refer Slide Time: 7:37)

2D Transformation

Rotation

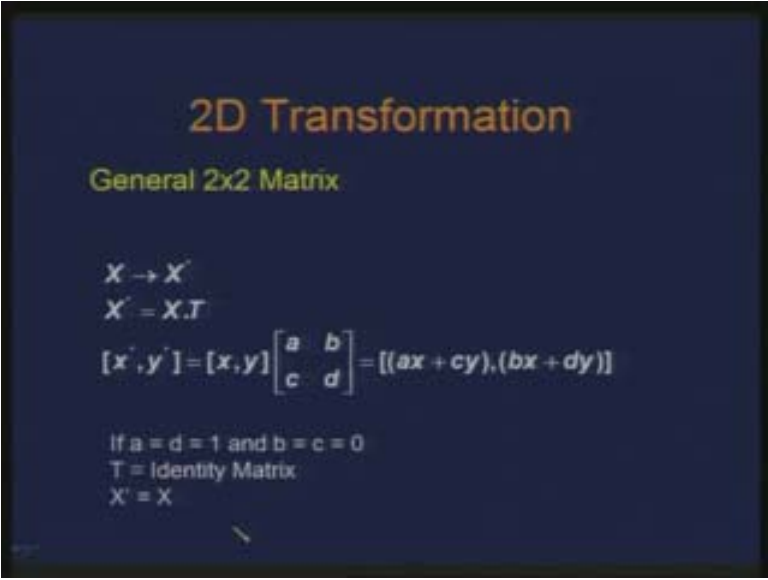
$P(x, y) \rightarrow P'(x', y')$
 $x = r \cos(\phi); y = r \sin(\phi)$
 $x' = r \cos(\phi + \theta)$
 $y' = r \sin(\phi + \theta)$
 $x' = x \cos(\theta) - y \sin(\theta)$
 $y' = x \sin(\theta) + y \cos(\theta)$
 $[x', y'] = [x, y] \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$
 $P' = P.R_\theta$



The next transformation for our study is rotation. Here again we observe the transformation from P to P prime. So here this point P is located which is after a rotation about the origin it goes to P prime. Again we are considering the rotation about the origin.

Again there could be a convention which would give me the rotation as a positive rotation when I go in counter clockwise direction it is a matter of convention. So let us consider these points to be in polar coordinate system. We are identifying in terms of the radius as the distance from the origin and the angle. then I can represent X as $r \cos \phi$ Y as $r \sin \phi$ then the new point P prime after the transformation having undergone a rotation by an angle theta would be given by this $r \cos \phi + \theta$ and y prime as $r \sin \phi + \theta$. Then I can expand this terms and substitute the term for $r \cos \phi$ and $r \sin \phi$ as X and y and eventually get X prime and y prime in terms of only theta, terms involving theta and (x, y) value. Once again I can write this transform point X prime Y prime as some sort of a matrix multiplication to the original point (x, y). And this matrix basically consists of cosine and sine terms. Therefore P prime is basically P multiplied by the matrix which I call it as a rotational matrix for the angle theta.

(Refer Slide Time: 10:15)



2D Transformation

General 2x2 Matrix

$$X \rightarrow X'$$

$$X' = XT$$

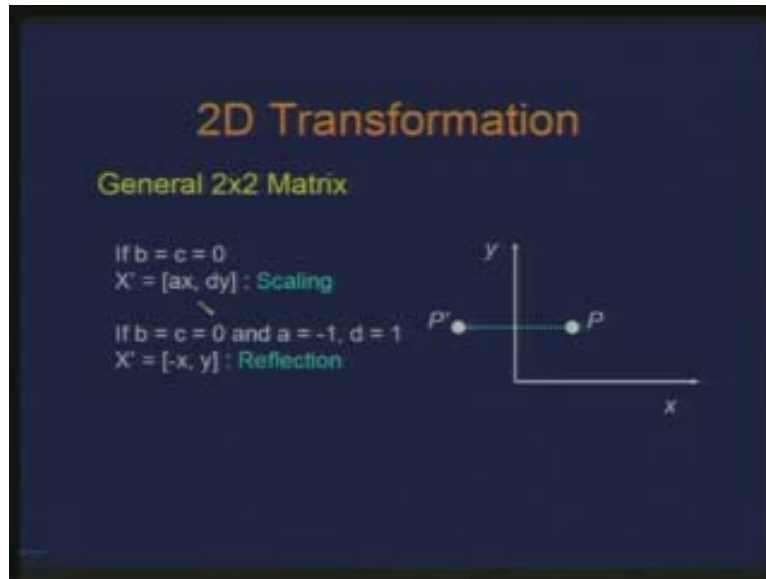
$$[x', y'] = [x, y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax + cy), (bx + dy)]$$

If $a = d = 1$ and $b = c = 0$
 $T = \text{Identity Matrix}$
 $X' = X$

Let us try to generalize this process of using matrix as a mechanism of performing transformation. So we have basically seen that scaling and rotation could be represented as matrix. Now if I take a point X going to X prime where X prime is nothing but a multiplication of a matrix T which I can consider to be a general 2 into 2 matrix having terms as a b c d then the transform point is given by ax plus cy and bx plus dy. And then it is actually the values of a b c d which would define the type of transformation you want to achieve.

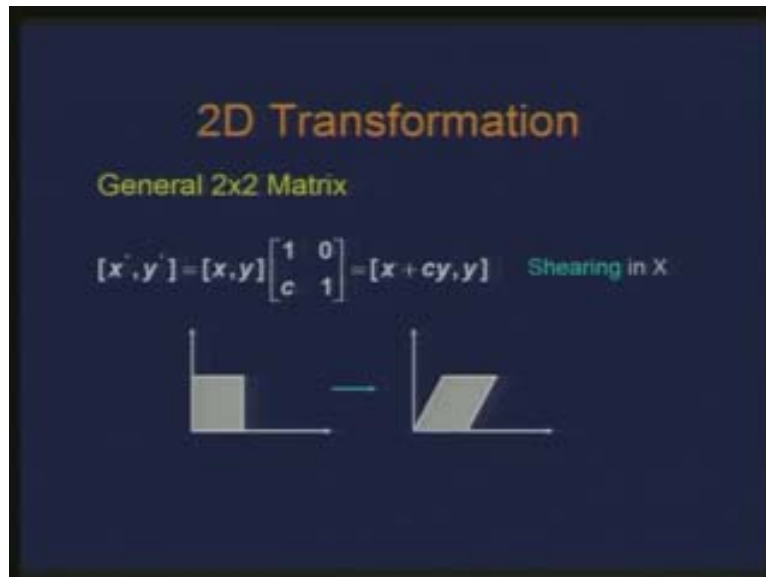
For instance, if I just have a is equal to d is equal to 1 and b is equal to c is equal to 0 and I am basically constructing that matters to be an identity matrix therefore nothing change. So everything boils down to something like a simple algebra using matrices.

(Refer Slide Time: 11:46)



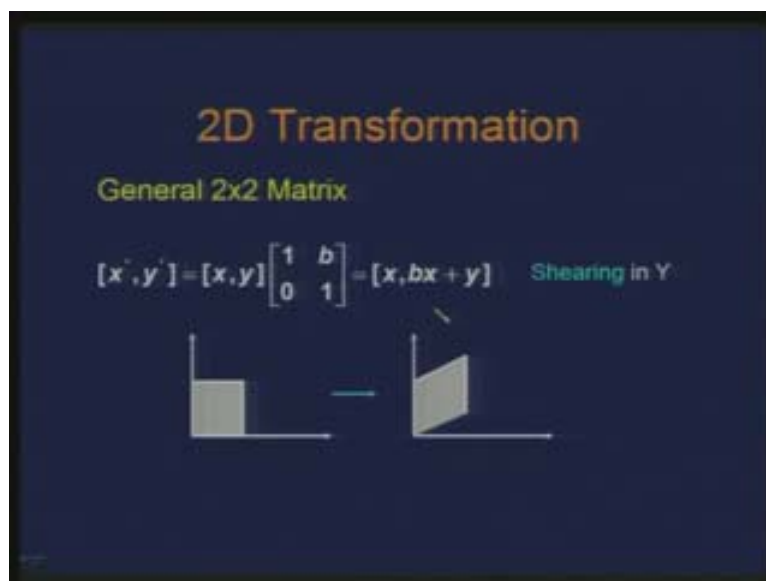
Now let us try to see that if I wanted to achieve scaling all I needed was basically only the diagonal terms and the off-diagonal terms as 0 meaning b and c are 0 then I get a scaling. And in fact another type of transformation called reflection is actually a special case of scaling. So there the half diagonal terms are 0, the diagonal terms are negative, one of them is negative then I am getting a reflection with respect to one of the axis and both the terms are minus 1 then I am basically getting reflection with respect to origin. So if I had both as minus 1 diagonal terms then the point will come somewhere here. Now basically we have figured out that this 2 into 2 general matrix can give as various types of transformations.

(Refer Slide Time: 13:09)



Let us consider another example where I considered the half diagonal terms. The diagonal terms are 1 and the off diagonal terms are non zero. So let us take one of the off diagonal terms here so what I get is x transforming to x plus Cy and y is the same as previous y so what I get as the transformation is some sort of a sharing of the shape. By shearing I mean that there is a shape like this and it becomes like this. It is some sort of a [s.....] 14:00 stretching and I call this as shearing in X because X is getting changed.

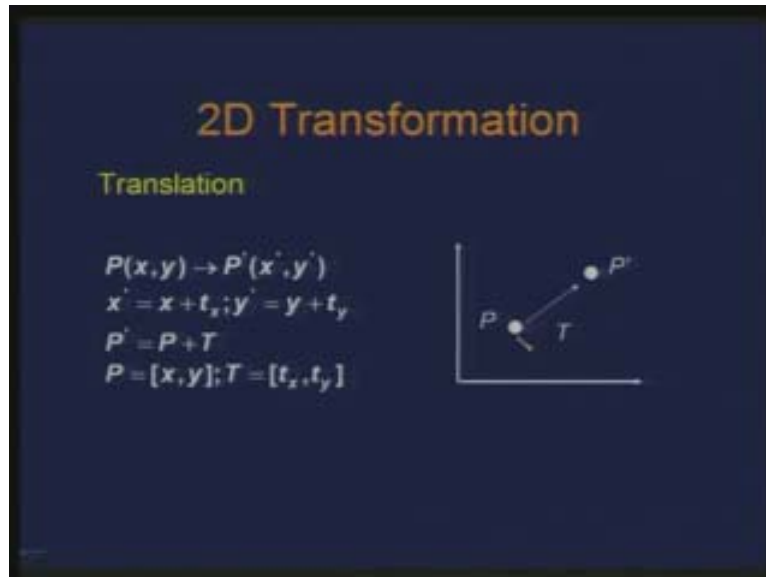
(Refer Slide Time: 14:19)



Similarly I can obtain shearing in Y where I would have the pff diagonal term b to be some non 0 therefore the y component is going to change as some function of X. So this

2 into 2 matrix which we have considered gives us various types of transformations. Now if I want to actually perform a transformation which is translation what I do is, basically there is a point P which is displaced to a point P prime with some offset displacement.

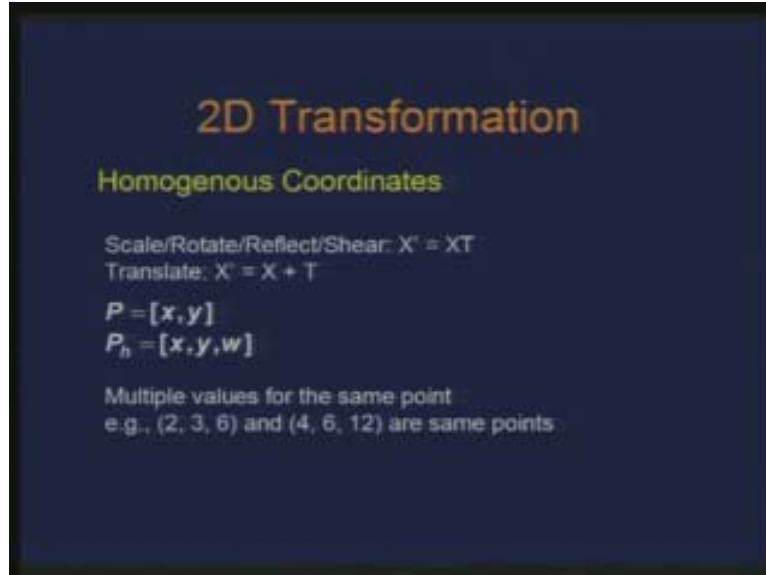
(Refer Slide Time: 15:00)



So if I have the point $P(x, y)$ which goes to P' prime having the coordinates as $(x' \text{ prime } y' \text{ prime})$ then what I am saying is that X' prime is actually X plus some offset in X which is given by t_x and Y' prime is y plus the offset displacement in y which is given by t_y . So in other words when I look at as an operation which is happening to the point p I find that P' prime is actually P plus T where T is a displacement vector and it is not the transformation matrix it is a displacement vector and I achieve this P' prime.

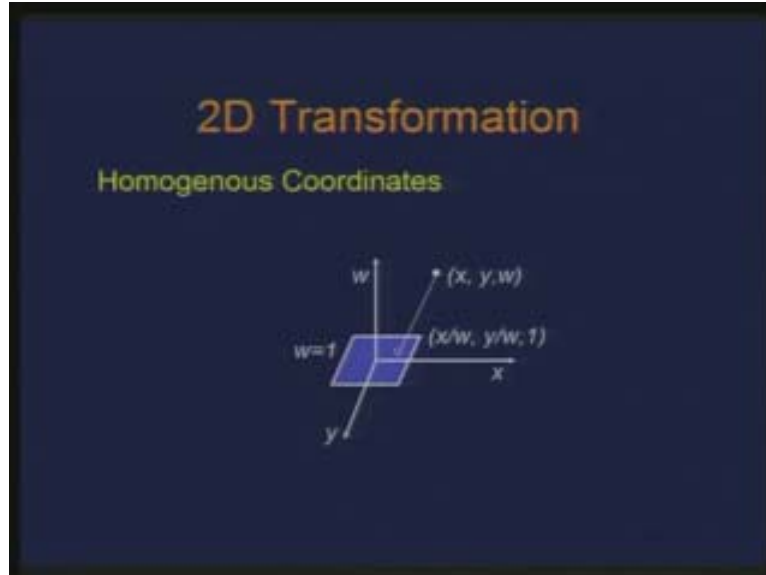
Now there is some kind of a problem here because basically I wanted to capture all the transformations in the form of a matrix which I could possibly do for many of the transformations but not for translations. Here there is no such matrix multiplication which will give me a translation. Some of the questions raised is, can I still be able to have a matrix representation incorporating translation. Basically I wanted to achieve just as a matrix multiplication for any transformation I want. And what I observe here is that translation is basically the **enormally** it does not fit into that. Now I want to devise a scheme by which I can still be able to do these transformations in the form of a matrix multiplication and for that we device what we call as the homogeneous coordinates.

(Refer Slide Time: 17:37)



Of course there are other properties and other characteristic of homogeneous coordinates but one of the features which we want to use at this point is to have some sort of a unifying mechanism. So if I say the matrix multiplication for the transformation was such as scaling, rotation, reflection, shearing as X' prime is equal to XT and the translation was happening as X' prime is equal to X plus T . So once again this T is different from that T it is not a matrix. Now what I do is I define the coordinates of T which would given as let say two coordinates X and Y with an additional coordinate which I call as W so there is this P which is now being represented in homogeneous coordinates as x, y, w so there is this third coordinates which I have basically added. Now with this I form a coordinate system where in fact I can have multiple values for the same point. For instance, the points with homogeneous coordinates as 2, 3, 6 and 4, 6, 12 are the same.

(Refer Side time 19:21)



Geometrically just to give an interpretation of this what actually happens here is that we have now added another coordinate to the point which is we call as W. So the point which was [ex..] 19:34 defined in a plane now has three coordinates x , y , w defined in a space given by x , y , and w . Now if I consider the Cartesian coordinates we basically use for all our geometrical purposes then it basically boils down into that where I assign the value 1 to this w which basically means that I divide x , y , w by this homogeneous coordinate w and I get X by w y by w and 1 which is nothing but a point on a plane w is equal to 1 basically corresponding to the line which is being found by all the multiple value representations of the point.

So basically the w part is being used as some sort of a scaling and all the points now form a ray or a line and the value of w which I choose gives me the projection of that line on to the plane w equal to whatever value I have. But for the purpose of our geometrical operations or the points which we represent in Cartesian coordinate system is given when w is equal to 1. So all these points basically which are on this line are the same points and it depends where I take the projection. Now having defined this as a new system of coordinates for the points let us try to investigate whether we can now unify the various transformations.

(Refer Slide Time: 22:12)

2D Transformation

Homogenous Coordinates

Unifying representation for transformation
Transformation matrix from 2x2 to 3x3

$$X \rightarrow X'$$
$$X' = X.T$$
$$[x', y', w] = [x, y, 1] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ l & m & 1 \end{bmatrix}$$
$$= [(ax + cy + l), (bx + dy + m), 1]$$

Basically the transformation matrix what we are considering now will go from 2 into 2 transformation matrix to 3 into 3 because now I have a definition of my points given by three coordinates with an additional homogeneous coordinate.

Once again if I look at the transformation which is happening to X prime through this transformation matrix T where this X is now given as x, y and 1 that is where we define all the Cartesian coordinates which in turn gives me X prime y prime and w. And the matrix which I have here is a 3 into 3 matrix where I have this part the upper portion of the matrix as the same as 2 into 2 matrix which I had earlier use and there are these terms l and m and deliberately I have put these 0 0 because the interpretation of those non 0 values we are discarding for the moment but there could also be some other values here. So a transform point basically is given by this.

(Refer Slide Time: 24:00)

2D Transformation

Homogenous Coordinates

Unifying representation for transformation
Transformation matrix from 2x2 to 3x3

$$X \rightarrow X'$$
$$X' = X.T$$
$$[x', y', w] = [x, y, 1] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ l & m & 1 \end{bmatrix}$$
$$= [(ax + cy + l), (bx + dy + m), 1]$$

Can we derive the various transformations we have seen through this? If I had translation, translation would basically mean that if I have the translation vector given by l and m where I place the translation component to the matrix and the rest of the matrix look like this then I am basically getting X prime and y prime values given as just the displaced values of X and y which is X plus l and y plus m. Therefore if I was just to perform the translation this is how my 3 into 3 matrix would look like. Therefore now I can perform the operation of translation as just a matrix multiplication.

(Refer Slide Time: 25:07)

2D Transformation

Homogenous Coordinates

Scaling/Rotation/Shear

$$T = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly, if I was to perform scaling or rotation or shear this 2 into 2 general matrix which I had considered earlier I would just place it there and there would not be any translation terms here so I put 0 0 here. And then again just the multiplication of this matrix T would give me the required transformation. So we have basically devised a scheme by which we can obtain the transformation in terms of matrix multiplication including translation. Now, if I have to perform successive application of transformation, that is we are considering one type of transformation.

(Refer Slide Time: 26:14)

2D Transformation

Homogenous Coordinates

Successive translations: $T_1 = (l_1, m_1)$, $T_2 = (l_2, m_2)$

After T_1

$$X' = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & m_1 & 1 \end{bmatrix}$$

After T_1 and T_2

$$X'' = X' \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_2 & m_2 & 1 \end{bmatrix}$$

$$X'' = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & m_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_2 & m_2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 + l_2 & m_1 + m_2 & 1 \end{bmatrix}$$

Successive translations are **additive**

Hence I am basically considering successive translations so the first translation I defined as a translation T_1 given by l_1 m_1 , the second translation I defined as T_2 given by l_2 m_2 then what happens in terms of the matrices which I would apply for the two transformations is first I will get X prime which is obtained after applying T_1 the first translation which is given by this matrix here having the term l_1 and m_1 and then I apply the second translation to whatever I get here which is X prime given by this matrix containing the terms for l_2 and m_2 which gives me the result as X double prime. Now all I have to do is substitute what was X prime here which basically gives me this matrix and I just multiply the two matrices for the two transformations and I get this. There should be X here actually.

(Refer Slide Time: 27:31)

2D Transformation

Homogenous Coordinates

Successive translations: $T_1 = (l_1, m_1)$, $T_2 = (l_2, m_2)$

After T_1

$$X' = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & m_1 & 1 \end{bmatrix}$$

After T_1 and T_2

$$X'' = X' \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_2 & m_2 & 1 \end{bmatrix}$$

$$X'' = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & m_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_2 & m_2 & 1 \end{bmatrix} = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 + l_2 & m_1 + m_2 & 1 \end{bmatrix}$$

Successive translations are **additive**

This is just the transformation matrix which needs to be multiplied to X, X multiplied by this. Therefore what you observe here is, with respect to the transformation matrix or the transformation which is happening is that you have these terms here which is l_1 plus l_2 and you have the term here m_1 plus m_2 which is now a counting for the two translations which in turn says that translations are additive in nature with respect to the transformation that is happening.

(Refer Slide Time: 28:19)

2D Transformation

Homogenous Coordinates

Successive scaling: $S_1 = (s_{x1}, s_{y1})$, $S_2 = (s_{x2}, s_{y2})$

After S_1

$$X' = X \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

After S_1 and S_2

$$X'' = X' \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X'' = X \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = X \begin{bmatrix} s_{x1}s_{x2} & 0 & 0 \\ 0 & s_{y1}s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Successive scaling is **multiplicative**

Now, if I want to perform successive scaling, just the same thing again. Here the scaling factors are given as S_{x1} and S_{y1} and for the second scaling I have S_{x2} and S_{y2} so I just do

the successive application of the transformation for each and what I see here is the resulting transformation which has the terms S_{x1} into S_{x2} and here S_{y1} into S_{y2} . Therefore I observe that successive scaling is multiplicative. Now what happens if I do successive rotation? It is additive, so we can see that now here.

(Refer Slide Time: 29:19)

2D Transformation

Homogenous Coordinates

Successive rotations: $R(\theta)$, $R(\phi)$

After $R(\theta)$

$$X^* = X \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

After $R(\theta)$ and $R(\phi)$

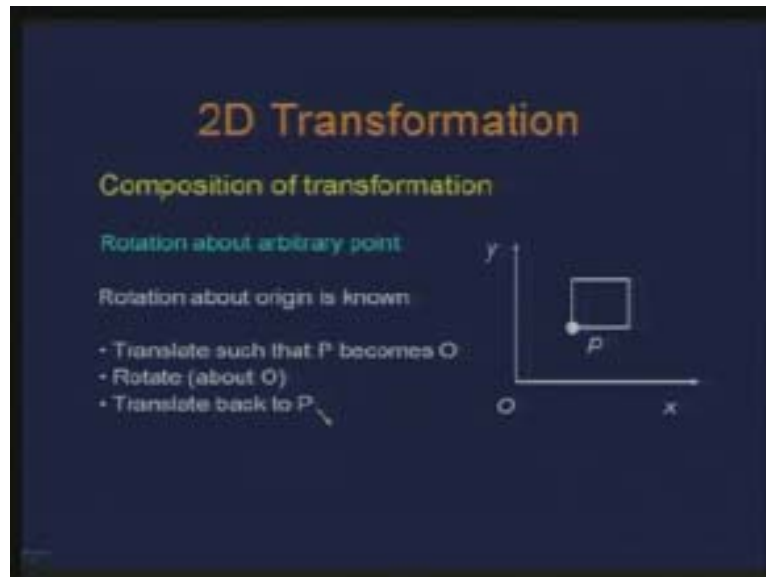
$$X'' = X^* \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X'' = X \begin{bmatrix} \cos(\theta + \phi) & \sin(\theta + \phi) & 0 \\ -\sin(\theta + \phi) & \cos(\theta + \phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Successive rotations are **additive**

I have a rotation given by an angle theta which I call as R theta and there is another rotation given by an angle phi which I call as Rphi. So, after the first rotation this is what will happen, after the second rotation this is what will happen and if I multiply by these two again this is the resulting transformation. This should be X double prime is equal to X multiplied by this matrix. So this is the resulting matrix, this is not really X double prime. So what you observe here is that successive application of rotations are additive in nature, you just add the angles by which they would be rotated.

(Refer Slide Time: 30:34)



Basically this was the first way to look at the composition of transformations which we were performing. So, there we were considering the same type of transformation but there could be composition of different type of transformations which we need to perform. There is an example where I have a figure here, I know how to do the rotation about the origin for that I have a matrix. But the problem is modified and states that now you have to perform rotation about the point P. So the question is can I perform rotation about P using the transformations which I have just seen. Since I know the rotation about origin I have to just basically map my problem for rotation about origin. The best thing is to make this point P origin, perform the rotation and then do the post rectification.

The first thing which is to be done is you translate this figure such that P becomes the origin O, you perform the rotation as you know the matrix for that and then translate back to P.

(Refer Slide Time: 32:38)


2D Transformation

Composition of transformation

Rotation about arbitrary point

Rotation about origin is known

- Translate such that P becomes O


$$X' = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -l & -m & 1 \end{bmatrix}$$

So, this is what happens when we do the translation, this figure gets translated to this and the necessary matrix which I would need is something like this. If I define this offset translation by l and m then I am forming a matrix of translation given by this.

(Refer Slide Time: 33:11)


2D Transformation

Composition of transformation

Rotation about arbitrary point

Rotation about origin is known

- Rotate about O


$$X' = X \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore these are minus l and minus m and then I do the rotation about this origin because I know what the transformation is and this is the transformation I require for performing the rotation about region and then I take it back for which I would need another translation and the transformation of that transition would look like this.

(Refer Slide Time: 33:43)

2D Transformation

Composition of transformation

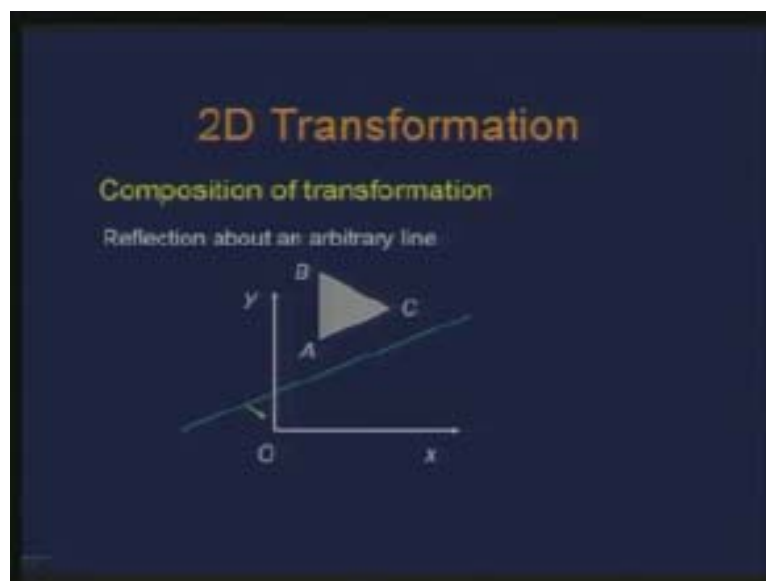
Composite transformation

$$X_c = X \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -l & -m & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l & m & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ -l(\cos \theta - 1) + m \sin \theta & -m(\cos \theta - 1) - l \sin \theta & 1 \end{bmatrix}$$

So, as a result all I need to do is concatenate these transformations. Just do the concatenation of these transformations which would give me the final transformation as this. So again the X is missing here. So this is the final transformation. That is a very big advantage here of using matrices as a representation of transformation as actually used in most of the graphics library we have. Here is another example which would illustrate the composition of transformation.

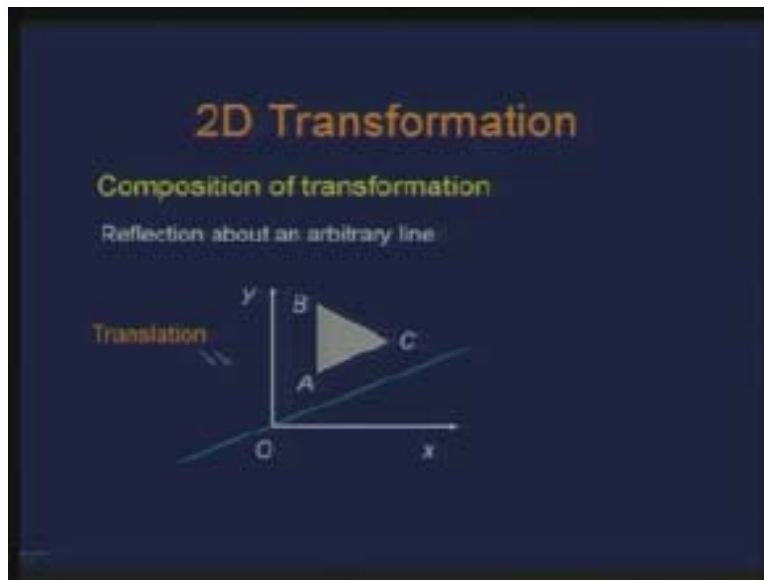
(Refer Slide Time: 34:43)



For instance, if I want to perform reflection about this line which is shown here in the blue color and this is the figure I have. Clearly I do not know reflection about an arbitrary

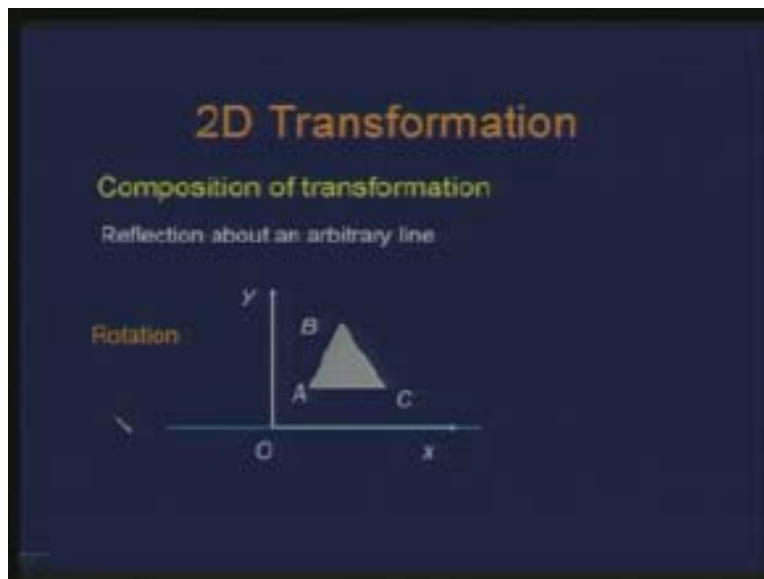
line; I do not know the matrix for that. But again by using the composition of various transformations I can obtain that matrix. And I know that if this line was one of the axes I could do the reflection, this much I know. Therefore I need to map this problem of reflecting about an arbitrary line to one of those where I could do the reflection with respect to one of the axis. So what we do is first basically translate the line past to the origin, rotate the line and proceed.

(Refer Slide Time: 36:19)



Therefore this is the translation which will basically make this line pass through the origin.

(Refer Slide Time: 36:28)



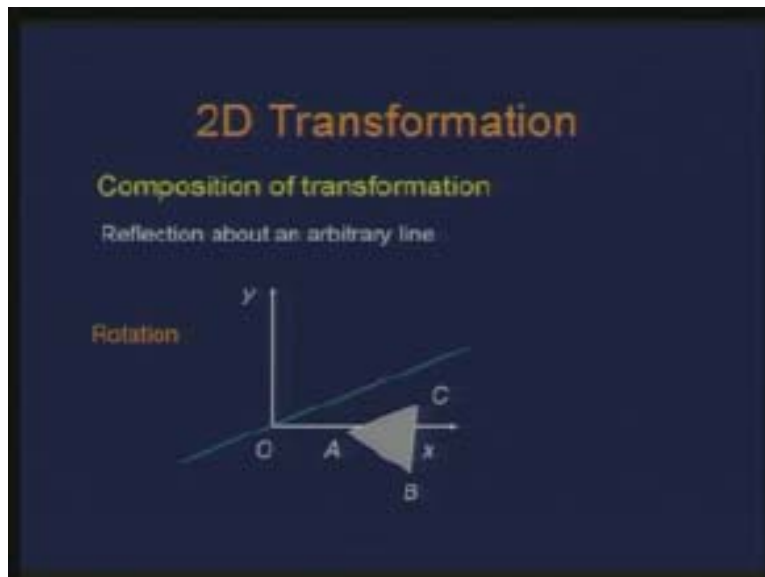
This would do the alignment of the line with respect to one of the axis; in this case it is the X axis.

(Refer Slide Time: 36:39)



This would do the necessary reflection with respect to the X axis or about X axis and then we do the reverse things.

(Refer Slide Time: 36:52)



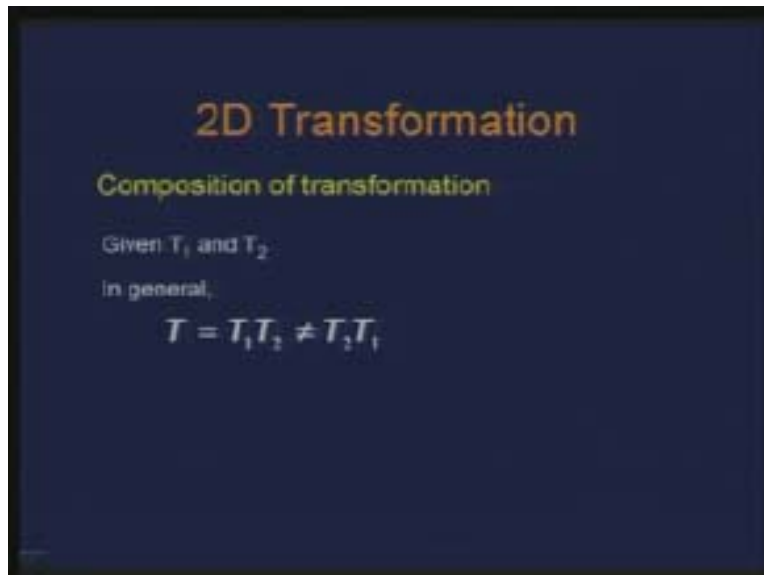
We rotate it back and then we translate it back.

(Refer Slide Time: 37:00)



This is the reflected figure we have for the given figure here. Just by using composition of transformation we can actually enrich the class of transformation.

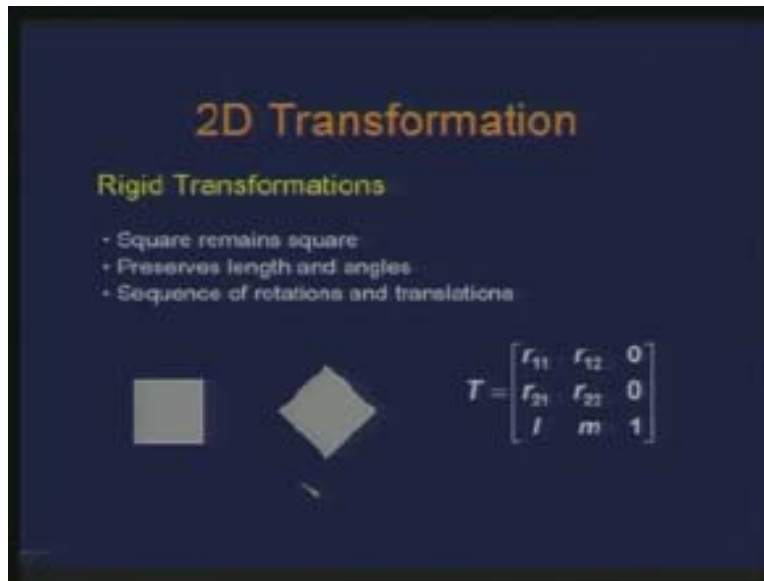
(Refer Slide Time: 37:29)



When we are talking about this composition of transformation there are words of warning. Given a transformation T_1 and another transformation T_2 in general the law of commutativity does not hold for matrix multiplication, they are not commutative. So $T_1 T_2$ is not equal to $T_2 T_1$. But in certain cases it would hold. What are the cases? If I have two scaling does not matter which one I do first to translation, it does not matter which one I do first rotations, which one I do first does not matter so they are commutative. But

if I have one rotation scaling in certain cases it could but not in all the cases or translation rotations would not hold this. One has to just be careful when we are using transformation as matrices then when you are trying to compose them then one has to be careful, you cannot just change the order of multiplication. Now let us see the class of transformation which we are looking at.

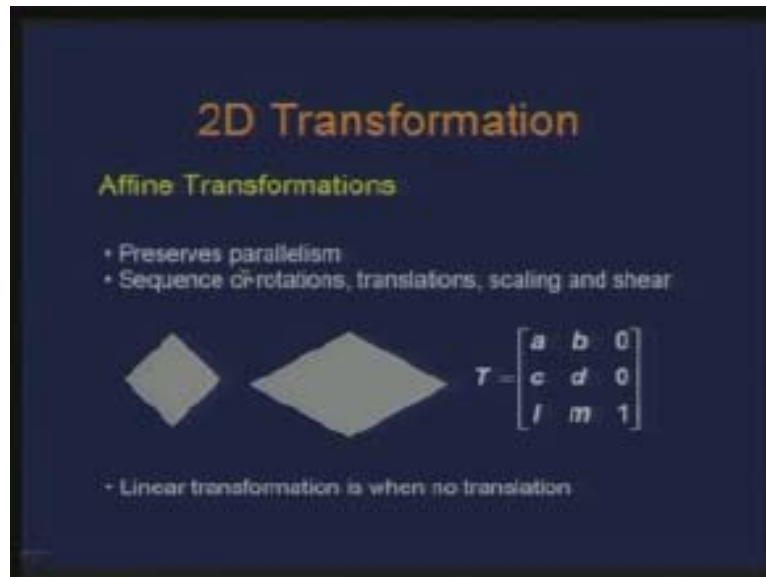
(Refer Slide Time: 39:45)



Just by seeing this matrix we can actually obtain different class of transformations. For instance, rigid transformations are those transformations where the shape is preserved meaning I preserve the length and angles. The shape and size are basically preserved in rigid transformations so square remains square. And that I would obtain if the component transformations are of the type rotations and translations which basically means that the 2 into 2 general matrix which I had considered a b c d has to be the rotation matrix then I can also have translation here which means I have the terms for l and m. This is how I can obtain the rigid transformations.

The other classes of transformation which are more general in nature are fine transformations. A fine transformation would basically mean that I have this general 2 into 2 matrix which I had used earlier plus translation.

(Refer Slide Time: 41:47)



What are the properties with respect to the shape it gives us? One thing is that it preserves parallelism. Parallel lines remain parallel after the transformation and the type of transformations which would give us the fine transformations basically would include the rotation, translation, scaling, and shear so all which is there as this 2 into 2 general matrix which we had considered plus the translation. And this 2 into 2 general matrix is actually a linear transformation. If I have to disregard the homogeneous coordinate system then X' basically given as X multiplied by the general 2 into 2 matrix maybe some A and then when I expand those terms it becomes a linear combination of the terms which is basically the linear transformation.

Therefore linear transformation is actually the subclass of a fine transformation when I do not have translation. Now, if you want to see the difference here, in the case of a linear transformation the origin is going to map to origin whereas in a fine transformation this may not be a case because there is a translation.

Hence this class of transformation a fine transformation is of a particular interest when one is trying to do a change of coordinate system. I define two axis X and Y and then I define another system of coordinate where I have some other axis instead of X and Y so using the fine transformation I can actually map from one coordinate system to another coordinate system. So the alignment of axis would basically be taken care by the linear transformations I have and the shift of the origin will be taken care by the translation. So now instead of having these as zero terms I can very well have non zero terms and that makes a general 3 into 3 matrix for transformation.

(Refer Slide Time: 45:10)



I basically have this 3 into 3 matrix where the last column now is changed to p q s. So instead of 1 now I have the term s and instead of having zero terms here I have non zero terms as p q. So if I had this matrix where the diagonal term a d both were one these off diagonal terms were 0s c l m and d p q and there was this s **stra** other than 1 what have you got? This is other than 1, so far we have seen s is equal to 1. Now I am giving you s other than 1 and just to simplify this situation the other terms are the off diagonal terms are 0, these are all 0, these are all 0 and a and d are both 1.

Therefore you have got a global scaling. Remember there is this X Y W and while getting back to the Cartesian coordinate system I would have to divide by this w. So, eventually the coordinates I have are X by w and y by w. So in this case this w will become s so I actually obtain a global scaling. And these p q terms are basically for the projective transformation. So the projective transformation is a super class of fine transformation where lines map to lines but parallelism may not hold, the parallel lines may not remain parallel after the projective transformation. This would have a better explanation and interpretation when we go to 3D transformations because that is way we will do lots of projection and so on.

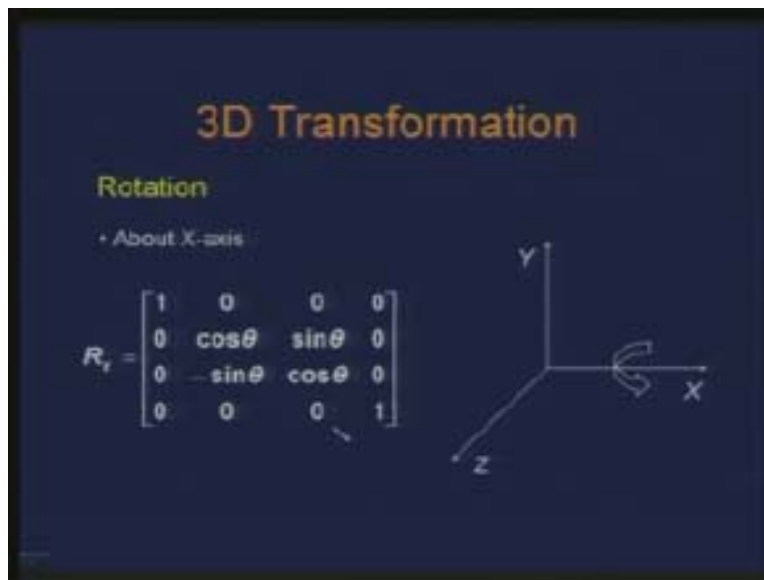
Now in fact having seen these 2D transformations extending them to 3D is straight forward. All you have to do is instead of using 3 into 3 matrix now you use 4 into 4 matrix. Once again we will work in homogeneous coordinate system. Now there is an additional dimension z so we have point X y z w. Examples: If I have to perform scaling now I can actually differentiate between local scaling and global scaling. So here is the matrix for local scaling. Local means individual to each axis.

(Refer Slide Time: 49:06)



So, s_x s_y and s_z are the respective scaling factors and this last term is 1.

(Refer Slide Time: 49:19)

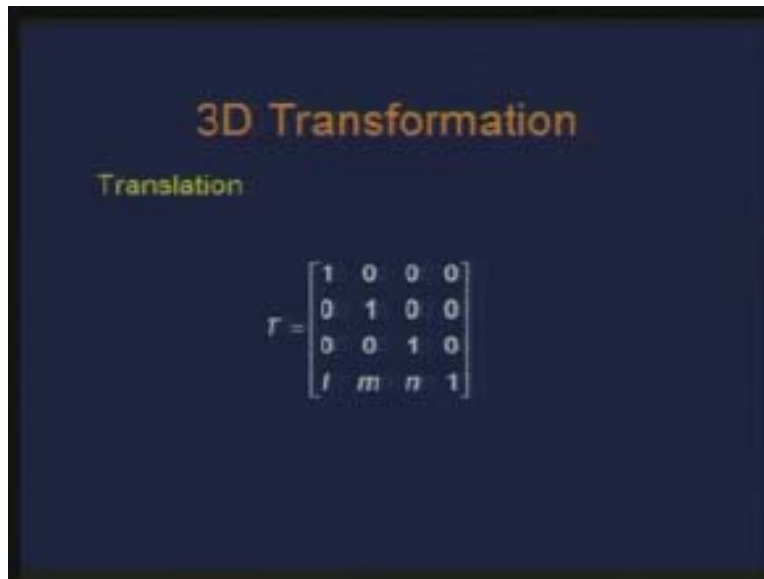


Similarly, now if I have to do a rotation, there the rotation was basically about the origin and implicitly about an axis which was perpendicular to the plane where that figure is. Now, when you talk about rotation in 3D basically I am going to talk about rotation about an individual axis.

For instance, the rotation about the X axis would be this which basically means I am rotating about X axis and I can again have a convention that if I use my right hand

coordinate system then the thumb is the axis about which I am rotating then the rotation of these fingers is giving me the positive rotation. This is known as a positive rotation with respect to X axis. Similarly, using this convention I can get the rotation about y axis so this will be actually negative and this will be positive. That is in line with my convention and rotation about Z axis and so on.

(Refer Slide Time: 51:21)



Therefore similarly I can have translation matrix where I have an additional term of n. This is a straight forward extend and there is nothing great about it.

(Refer Slide Time: 51:39)



And the off diagonal terms basically give me the shear.

(Refer Slide Time: 51:51)



Now the whole transformation basically can be represented as a general 4 into 4 matrix which is a simple extension of general 3 into 3 matrix which I used in 2D. Maybe just for the convention sake I have been using the points which are basically represented as a row vector where I post multiply the row vector of my point. You can also use the column vector representation of the point where you would pre multiply. So this is just a matter of convention.

In some of the books you may find a switch where instead of using this post multiplication they will be using pre multiplication but that is just a convention. These are basically a class of geometrical transformation where the geometry is modified.