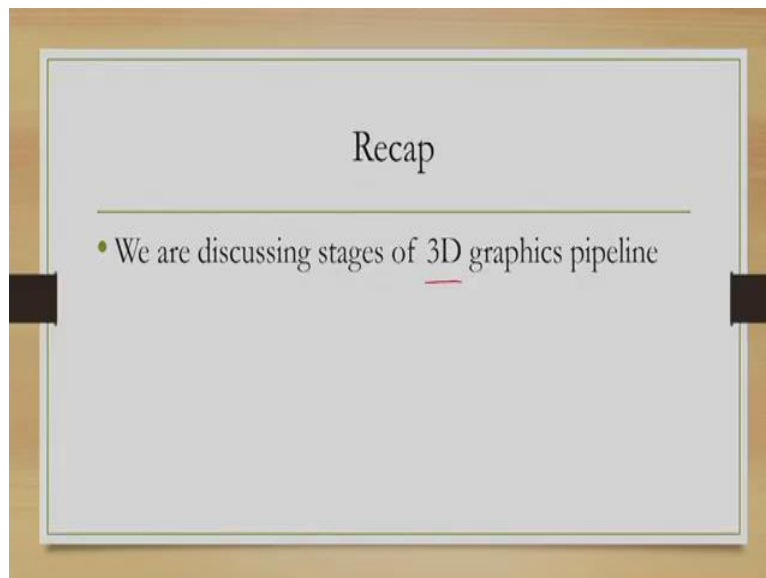


Computer Graphics
Professor Dr Samit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati
Lecture 10
Introduction to Modeling Transformations

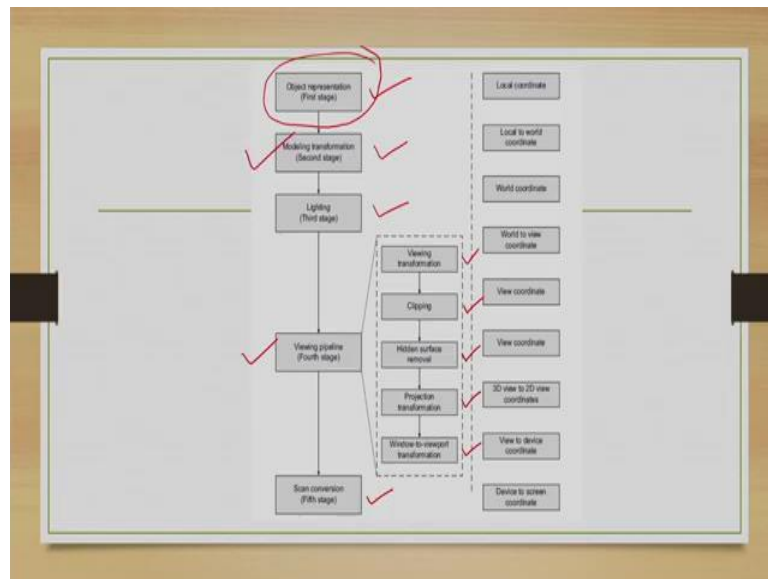
Hello and welcome to lecture number 10 in the course Computer Graphics. We were discussing about the graphics pipeline. To recollect, in computer graphics what we do, we generate 2D images on a computer screen and the process of generating these 2D images involves a set of stages, together these stages constitute the graphics pipeline. So, let us have a relook at the 3D graphics pipeline.

(Refer Slide Time: 1:15)



When we are talking of 3D graphics pipeline, just to refresh our memory, we are talking of creation of a 2D image from a 3D description of a scene.

(Refer Slide Time: 1:25)

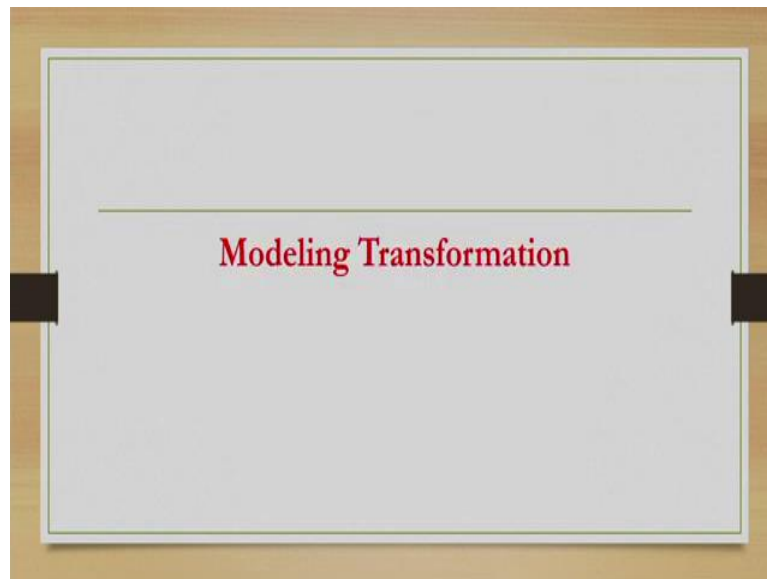


So, what are the stages that are part of the pipeline? First stage is object representation, second stage is modeling transformation, third stage is lighting or coloring, fourth stage is viewing pipeline. Now, note here that we are calling this stage as a pipeline itself, that means it consists of some more sub stages. There are 5 such sub stages viewing transformation, clipping, hidden surface removal, projection transformation and window to viewport transformation. And the final stage is scan conversion, also called rendering.

Among these stages, in the previous few lectures, we have covered the first stage, that is object representation. So, we now know how we can represent the objects that constitute a scene. Again, if I may repeat, the idea of computer graphics is to convert this representation to a sequence of 0s and 1s and that conversion takes place through the execution of these stages. The very first stage is object representation which we have already discussed.

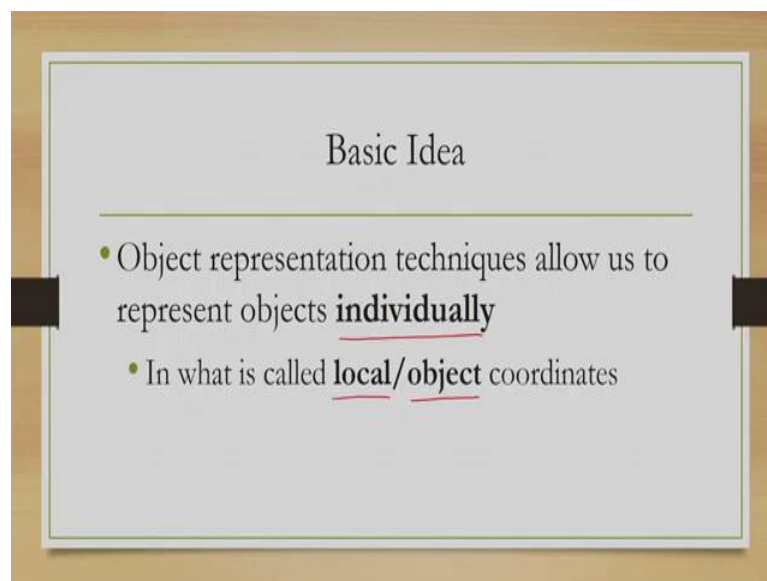
Today, we are going to start discussion on the second stage, that is modeling, also called geometric transformation.

(Refer Slide Time: 3:10)



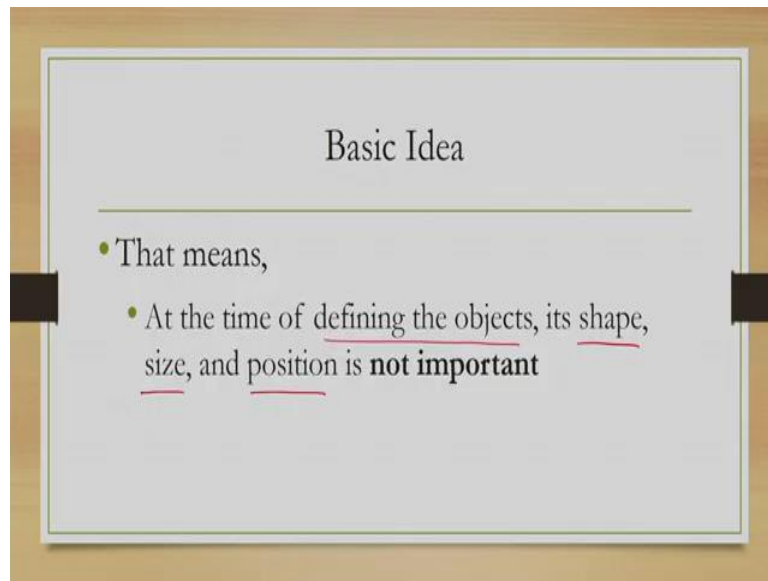
So, what is there in this stage?

(Refer Slide Time: 3:14)



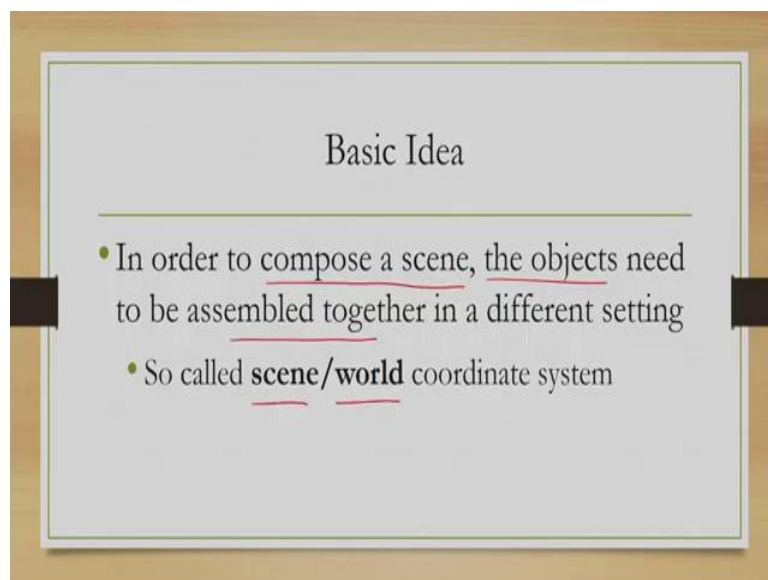
Now, when we talked about representing objects in the earlier lectures, what we implicitly referred to is that the objects were represented individually. Now, when we are defining those objects individually, we are implicitly using one coordinate system that is typically called local or object coordinate system. So, for each object definition we are having a local or object coordinate. Now, within this coordinate system we are defining the object.

(Refer Slide Time: 3:57)



What that means? That means, at the time of defining objects, we are actually not bothering too much about the object shape, size and position relative to other objects. So, we have defined an object, but in essence there may be multiple objects where this particular object may have a relative size or relative position or a relative orientation. But when we are defining the object in its own local coordinate, we are not paying too much attention to those factors.

(Refer Slide Time: 4:42)

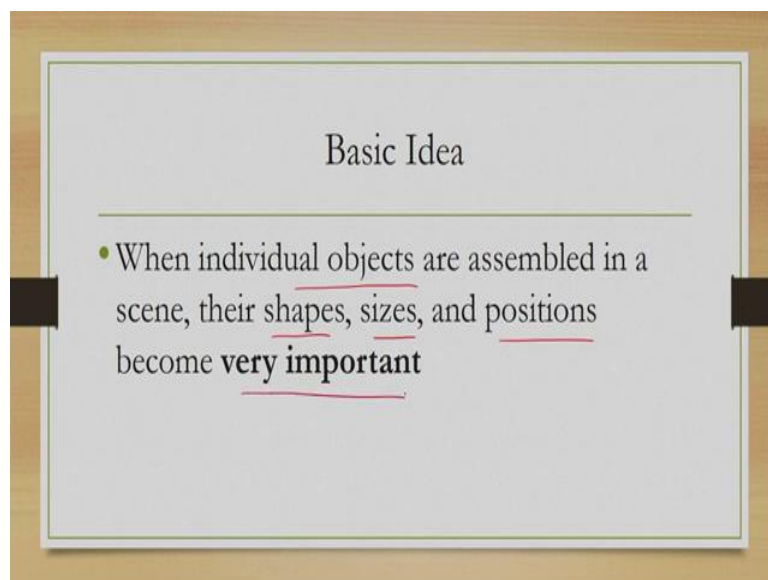


But, when we are going to compose a scene, which constitutes all the objects that we have defined, the objects need to be assembled together. Now, when we are talking of assembling the objects, what it means? It means that the objects should be put in a way it says that the

overall scene becomes perceptible. So, the object shape, size, orientation now is very important. So, when we are defining objects in its own coordinate those things are not important, but when we are assembling them, it becomes important.

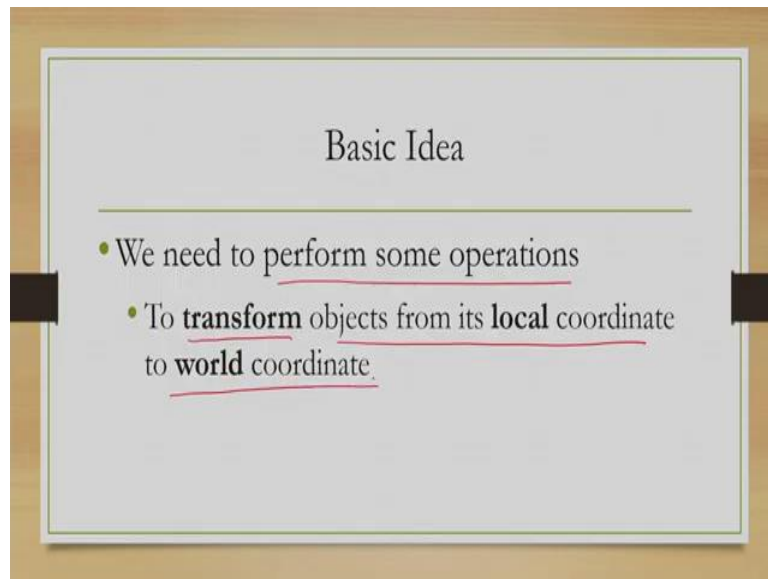
So, when we are trying to compose a scene by taking care of the relative shape, size, orientation and position of the objects with respect to other objects, we are again implicitly assuming another coordinate system, that is typically called the scene or more popularly the world coordinate system. So, earlier we dealt with local or object coordinate system, now, we have to deal with another coordinate system that is popularly known as the world coordinate system.

(Refer Slide Time: 6:06)



And as I said, in world coordinate system, the shapes, sizes, positions, orientations of these individual objects needs to be taken care of, so that we can construct the scene. So, those become very important now, in the world coordinate system.

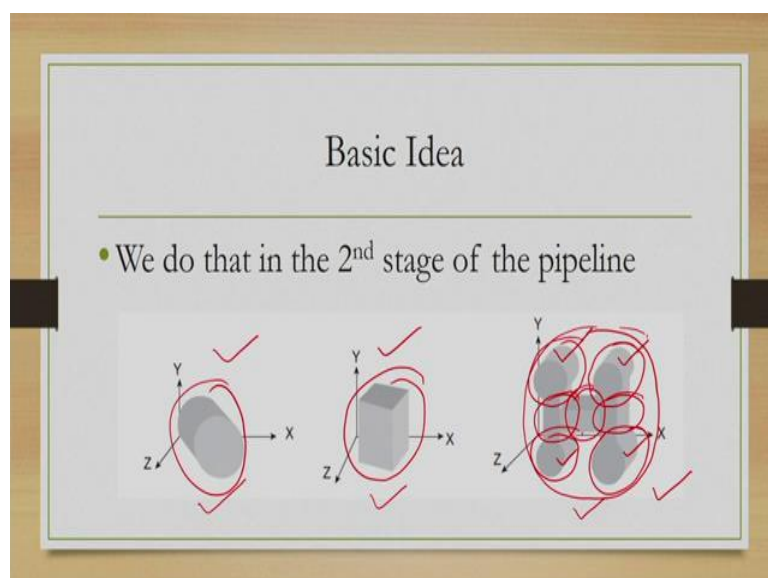
(Refer Slide Time: 6:29)



How we can do that? So, earlier we have defined objects in their own coordinate system without bothering about the relative shape, size, position, orientation, etc. Now, we are going to assemble them together in the world coordinate scene. And now, we have to think about those shapes, sizes, partitions orientations, so that the scene becomes perceptible. We can do that by applying some operations, by performing some operations.

These operations will transform the objects from its local coordinate to world coordinate. So, in order to create the scene by assembling the objects together, which in turn are defined in their own coordinate system, what we need to do, we need to perform some operations to transform the objects from their local coordinate description to world coordinate description.

(Refer Slide Time: 7:41)



Now, these operations or the transformation takes place in the second stage of the graphics pipeline. Let us see one example. Here we have 2 objects, this is object 1 on the leftmost figure and object 2 in the middle figure. Now, we want to create this object shown in the right hand figure. So, as you can see in this object in this overall object what we have, we have these cylinders, how many instances 1, 2, 3, 4 and the other shape how many instances 1, 2 and 3.

So, we have 4 instances of this object and 3 instances of this object. And the way these objects are defined in their own coordinate system are not the same as the way they are defined in this scene, which is on the right hand side. Here as you can see, here the orientation is different, orientation is different size is also different in all the four instances. Same is true for the other objects, the instances of the other object.

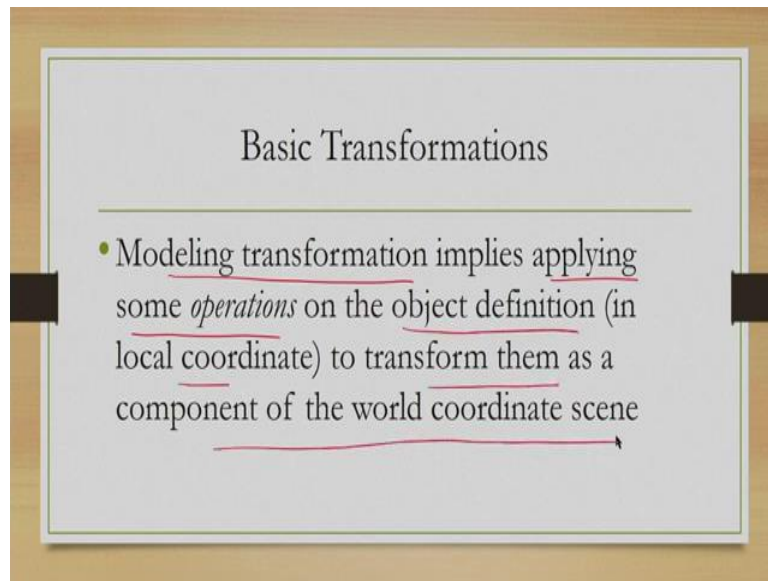
So, these coordinates where the objects are originally defined are the local or object coordinates. The coordinate here represented by the principal axis X, Y, Z in the right hand figure is the world coordinate. Here we are assembling multiple instances of the original object definitions to construct the overall object. And as you can see here, in order to do that, it is very important that the objects are put into proper place, in proper orientation and proper size. So, that is what is the job of the transformation that takes place in the second stage.

(Refer Slide Time: 10:10)



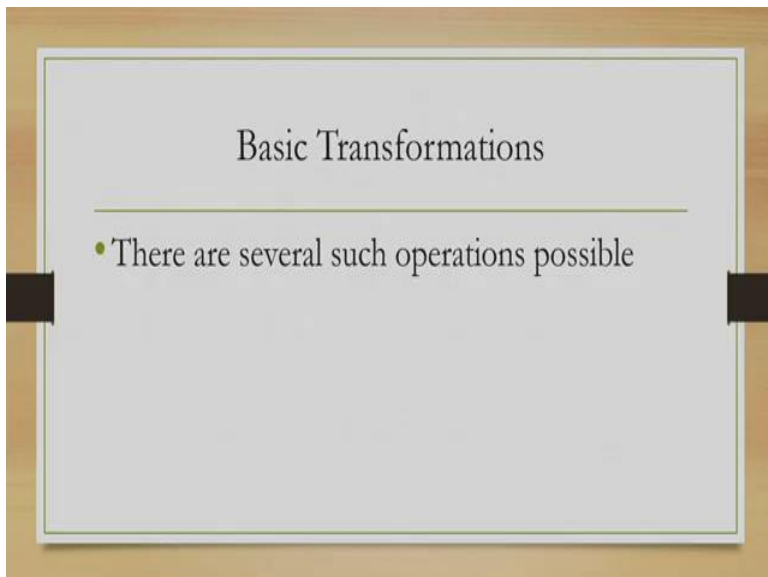
Now, since these transformations change some geometric properties or takes place on the geometry of the object definition, so we call these as geometric transformations. Also it is known as modeling transformations.

(Refer Slide Time: 10:38)



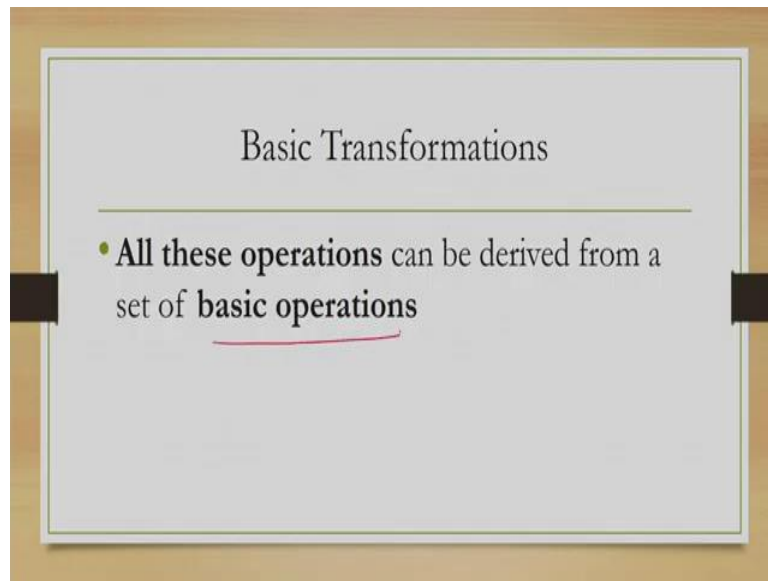
Now, these modeling transformations imply applying some operations on the object definition in local coordinate to transform them as a component of the world coordinate scene. So, this is what we can more formally talk about the modeling transformation, that is applying some operations on the object definition to transform them as a component in the world coordinate scene.

(Refer Slide Time: 11:11)



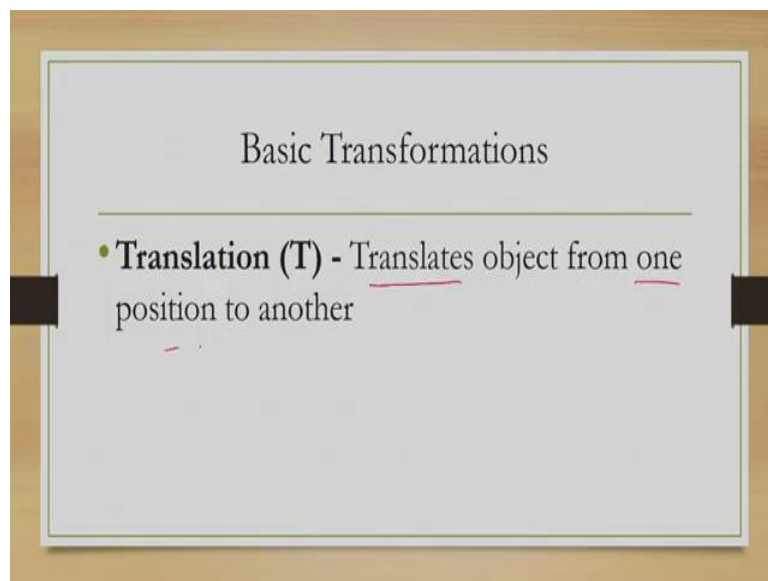
What are those operations, in fact, there can be many such operations. We will soon see what are those operations.

(Refer Slide Time: 11:23)



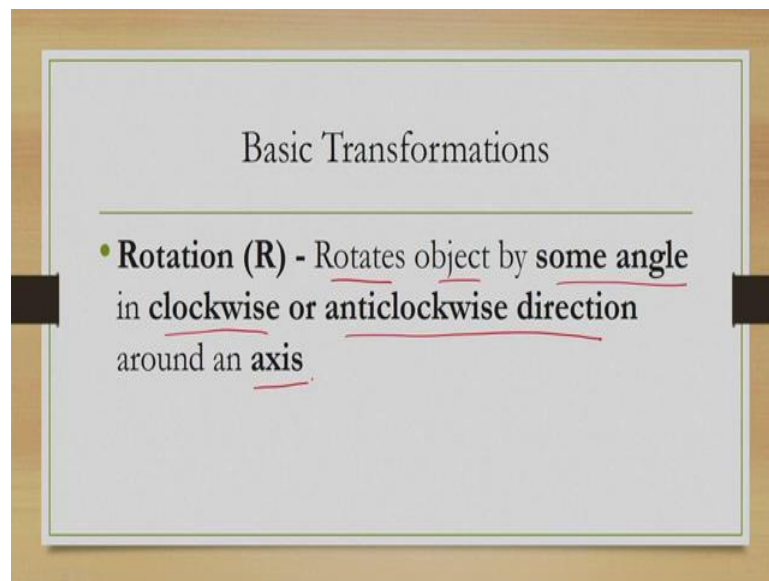
But all the operations can be derived from a set of basic operations. So, although we can in principle apply many operations, but these operations can be thought of as derived from a set of basic operations.

(Refer Slide Time: 11:43)



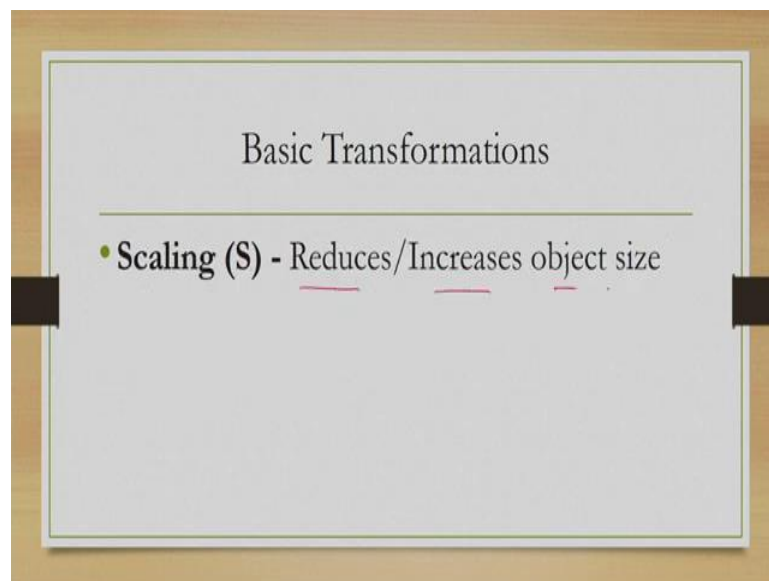
Now, let us have a look at those basic operations. There are actually 4 such basic operations, translation, rotation, scaling and shearing. Translation all of us know, what it does, it translates object from one position to another position.

(Refer Slide Time: 12:03)



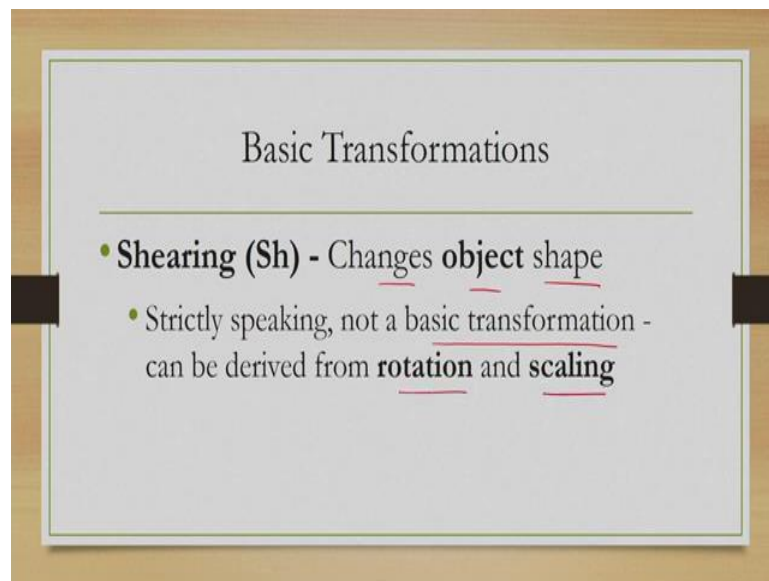
In rotation, what we do? We rotate the objects by some angle either in clockwise or anti clockwise direction around some axis.

(Refer Slide Time: 12:22)



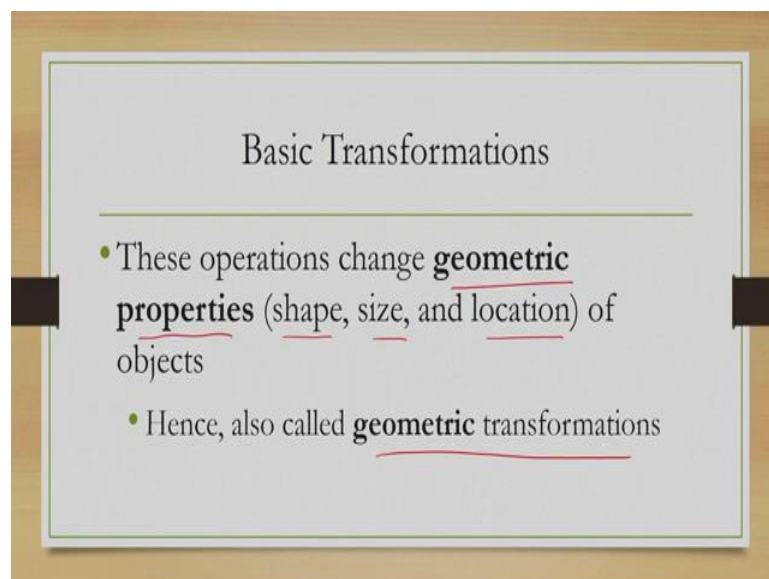
With scaling what we can do? We can reduce or increase the object size.

(Refer Slide Time: 12:30)



And finally, with shearing, we can change the shape of the object. It may be noted here that shearing is in a stricter sense, not a basic transformation and it can be derived as a composition of rotation and scaling. However, for simplicity, we will assume that this is a basic transformation and we will create it accordingly. So, then, let us recap. So, we have 4 basic transformations, translation, rotation, scaling, and shearing. Among them, shearing changes the shape of the object, scaling changes the size of the object, translation and rotation changes the position and orientation of the object.

(Refer Slide Time: 13:24)



Now, I am assuming that you have some idea of these operations and you may be knowing that these operations change the geometric properties of the objects in terms of changing their

shape, size and location. Since, that is the case, so we call these transformations as geometric transformers. So we can call the operations performed in the second stage as either modeling transformation or geometric transformation. Now, let us go into a little deeper discussion on each of these transformations. Let us start with translation.

(Refer Slide Time: 14:10)



What happens in translation?

(Refer Slide Time: 14:11)

A presentation slide with a light gray background and a thin black border. The word "Translation" is centered at the top. Below it, there are three bullet points. To the right of the text is a diagram of a 2D Cartesian coordinate system. The origin is at the bottom-left. A point $P(x, y)$ is marked with a red dot. A dashed point $P'(x', y')$ is shown to its right and above. A horizontal arrow labeled t_x points from the vertical line of P to the vertical line of P' . A vertical arrow labeled t_y points from the horizontal line of P to the horizontal line of P' .

Translation

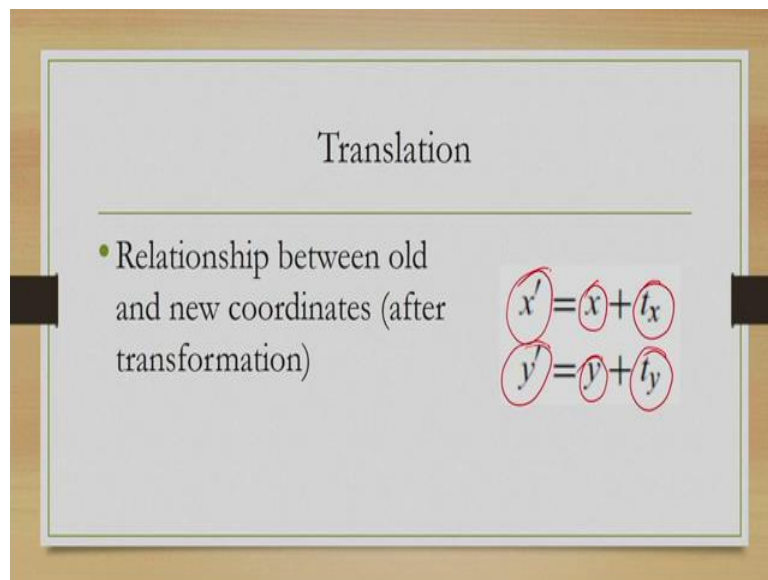
- Simplest of all
- To translate a point $p(x, y)$ to a new location $p'(x', y')$
 - Displace point by t_x and t_y along the X and Y directions

As you can see, suppose we have an original point here in this reference frame, which is denoted by P with coordinate x and y. Through translation we can reposition the point to P dashed with new coordinates x dashed and y dashed. So essentially we are displacing this

point to another point by an amount t_x and t_y to get the new point and this displacement takes place along the x and y direction.

So using this knowledge we can actually derive the new coordinate with respect to the old coordinate. What it will look like?

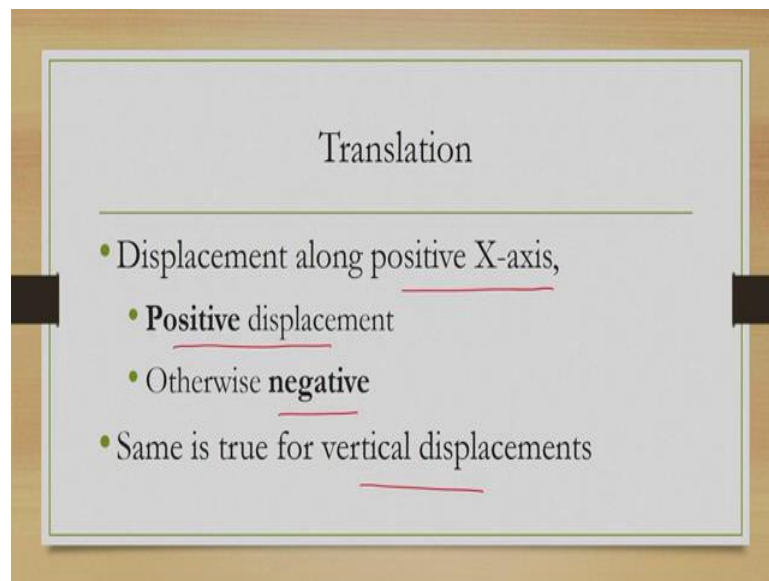
(Refer Slide Time: 15:06)



The slide is titled "Translation" and contains a bullet point: "Relationship between old and new coordinates (after transformation)". To the right of the text are two equations: $x' = x + t_x$ and $y' = y + t_y$. In the equations, the variables x' , x , t_x , y' , y , and t_y are each circled in red.

So the new coordinate, the new x coordinate will be x plus the displacement amount along the x direction and the new y coordinate will be the original y coordinate plus the displacement amount along the y direction. So these are simple derivations, and simple to formulate. And these are the relationships between the new and the old x and y coordinates of the points.

(Refer Slide Time: 15:41)



Now, these displacements can be thought of in different ways. So if we are moving along positive x axis or positive y axis, then we call it positive displacement. If we are moving along negative x axis or negative y axis, we call it negative displacement. So the sign of the amount t_x or t_y will be different for positive or negative displacements.

(Refer Slide Time: 16:21)



Now, let us shift our attention to rotation.

(Refer Slide Time: 16:25)

Rotation

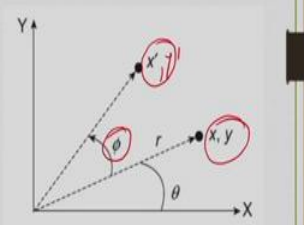
- Involves angular displacement
 - Point moves from one position to another on a circular track about some axis

Now, in case of rotation, we do not have horizontal or vertical displacements, instead we have angular displacement. In other words, the point moves from one position to another on a circular track about some axis. So, here, we are having angular displacement and the point is moving around some axis.

(Refer Slide Time: 16:57)

Rotation

- We assume rotation around **Z-axis** counterclockwise by an angle ϕ



We follow some convention typically, that is if the movement is counterclockwise, then it is positive angle of rotation. So consider this example here, we have the original point here. And now, we are having the point after rotation, it should be denoted by x dashed, y dashed and the rotation angle is ϕ . Now, since we are moving in the counterclockwise direction, we are calling it positive rotation angle.

If we are moving in clockwise direction, then we typically consider that to be a negative rotation angle. That is one convention typically followed. And in case of 2D rotation, we typically assume that the rotation takes place around the Z axis. However, later on we will see for 3D rotation what are the conventions.

(Refer Slide Time: 18:13)

Rotation

- Relation between new and old coordinates

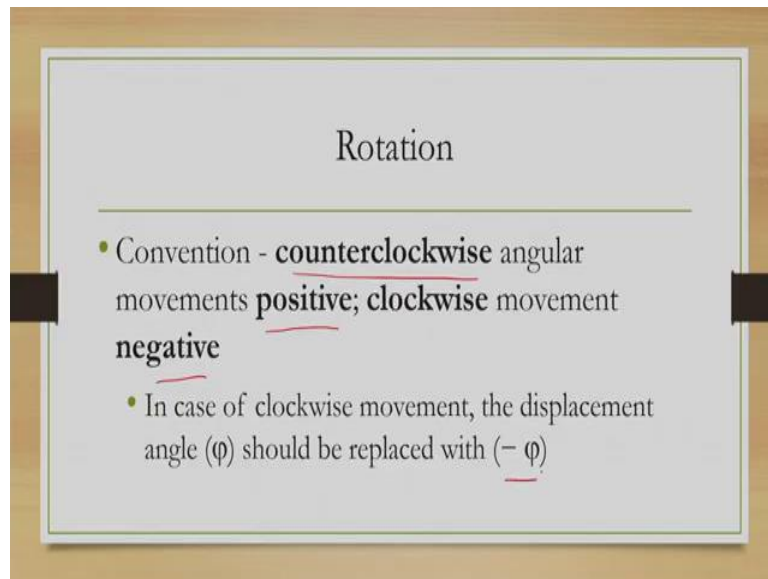
$$x = r \cos \theta, y = r \sin \theta$$

$$\left. \begin{aligned} x' &= r \cos (\theta + \phi) = r \cos \theta \cos \phi - r \sin \theta \sin \phi = x \cos \phi - y \sin \phi \\ y' &= r \sin (\theta + \phi) = r \sin \theta \cos \phi + r \cos \theta \sin \phi = x \sin \phi + y \cos \phi \end{aligned} \right\}$$

Now, let us try to derive the relationship between the new and old coordinates. So, the old coordinate is (x, y) , the new coordinate is (x', y') . Now, as you can see, we can represent x as $(r \cos \theta)$, now r is the radius of the circular track and θ is the angle between the x axis and the original point and y is $(r \sin \theta)$. Now, as we can see x' is if we draw a line like this, then we can represent x' as $\{(r \cos \theta) \cos \phi\}$. Now, if we expand then we will get $\{r \cos \theta \cos \phi - r \sin \theta \sin \phi\}$.

Since $(r \cos \theta)$ is x , so it is $(x \cos \phi)$ and since $(r \sin \theta)$ is y , so it will be $(y \sin \phi)$. So, similarly for y' , we can have a similar expression $\{x \sin \phi + y \cos \phi\}$. These two are relationship between the old coordinate of the point and the new coordinates of the point.

(Refer Slide Time: 19:44)



And as I already mentioned, counterclockwise angular movement is typically considered as positive, otherwise it is negative. Now, in case of negative angular movement, we change the sign of the angle of displacement. Instead of ϕ , we will use $-\phi$. That is the only change we will make.

(Refer Slide Time: 20:11)



So we have learned about translation and rotation. How we can apply these transformations to the points?

(Refer Slide Time: 20:23)



The way we derived, they applied to a point.

(Refer Slide Time: 20:33)



For an object we have many points. So single application to a point will not be sufficient. So what we need to do? We simply apply on all the points that make up the surface. Now, you may be thinking that that is impossible because there maybe infinite number of points on a surface. However, we can actually do that by applying the transformations to all vertices in a vertex list representation or all the control points for a spline surface.

So, when we are going to apply it, we essentially think of some representation, it can be a vertex list representation as in case of mesh representation, or can be a set of control points as

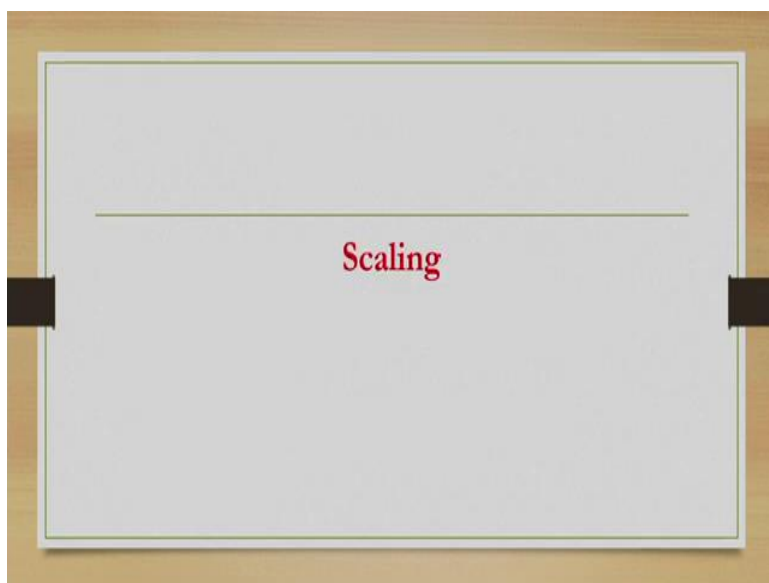
in case of spline representation, and we apply the transformations on each of these points are so that the entire object gets transformed.

(Refer Slide Time: 21:43)



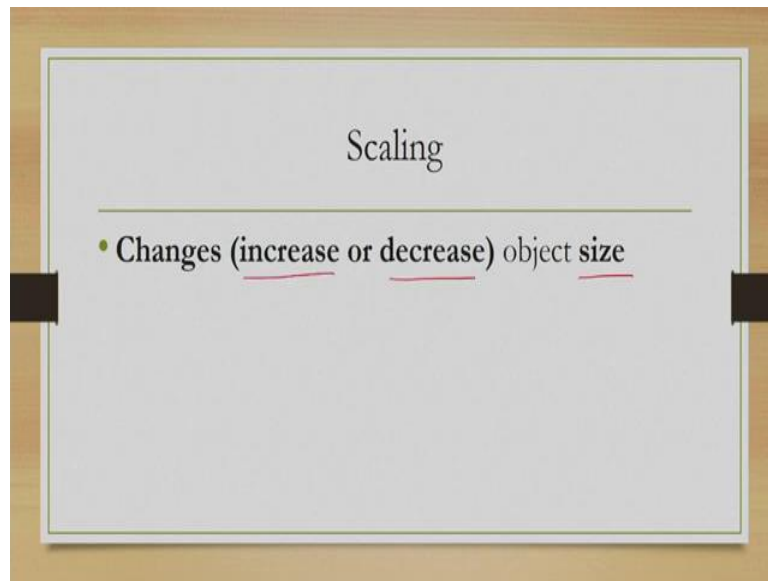
And as I have mentioned earlier that by applying in this way, we can change the object orientation using rotation and position using translation. So by applying rotation on all the vertices or all the control points, we can change the orientation of the object and by applying translation on all the vertices or all the control points, we can change the position.

(Refer Slide Time: 22:17)



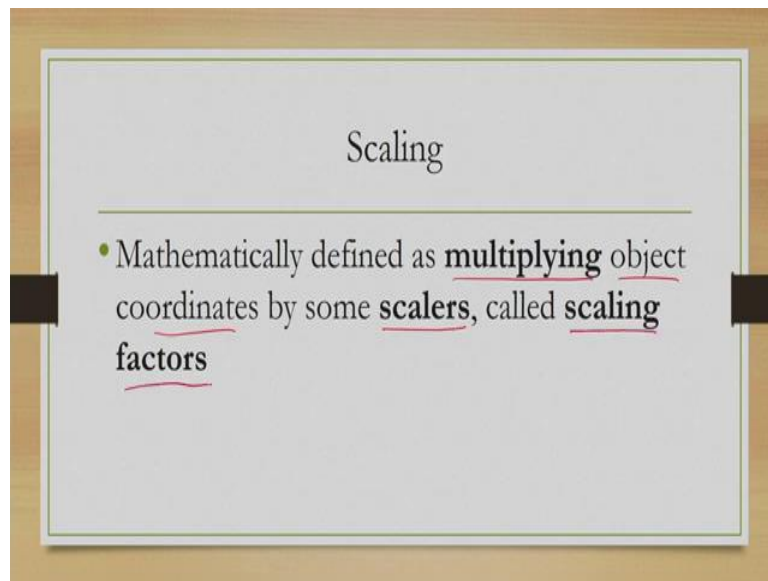
The third basic transformation is scaling. What happens in scaling?

(Refer Slide Time: 22:25)



It changes the size. Now, changes can take place in both ways, either it can decrease or it can increase. So both increase and decrease of the object size is possible with scaling.

(Refer Slide Time: 22:43)



Now, mathematically how scaling is defined? It is defined as an operation of multiplying object coordinates by some scalar quantity. Now, these scalar quantities are known as scaling factors. So essentially we are multiplying scaling factors to the coordinate values to scale up or down the objects. Scale up means increasing the size of the object, scale down means decreasing the size of the object.

(Refer Slide Time: 23:27)

Scaling

- Given a point $p(x, y)$, we can scale it by a factor s_x along X-axis (direction) and s_y along Y-axis (direction) to get the new point $p'(x', y')$

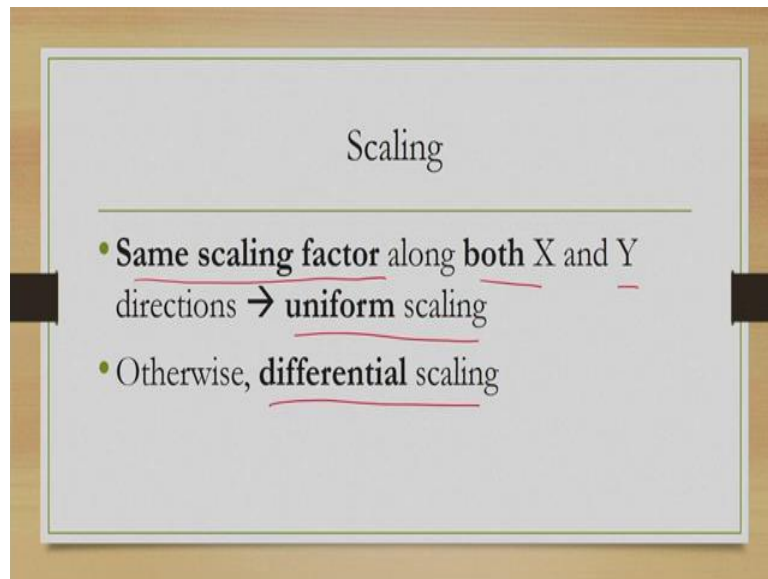
$$x' = s_x \cdot x$$
$$y' = s_y \cdot y$$

So, at the level of a point, how we can understand it? So, given a point P, we simply multiply the x coordinate with a scale factor along the x direction and a scaling factor along the y direction, we multiply this to the y coordinate to get the new point (x', y'). So, the relationship between the old and new coordinates will look something like this. So the new coordinate x' can be represented in terms of x in this way, and the new coordinate y' can be represented in terms of y in this way, where s_x and s_y are the two scaling factors along the corresponding x directions.

For example, here we have one object and we are using scaling factor along x direction to be one third and scaling factor along y direction to be half. Now, if I multiply these scaling factors to the x and y coordinates of the vertices, I will get 4 new vertices as shown here in this right-hand figure. Now these vertices together will represent the object. Since the scaling factors are less than 1, that means we are scaling it down or decreasing the size.

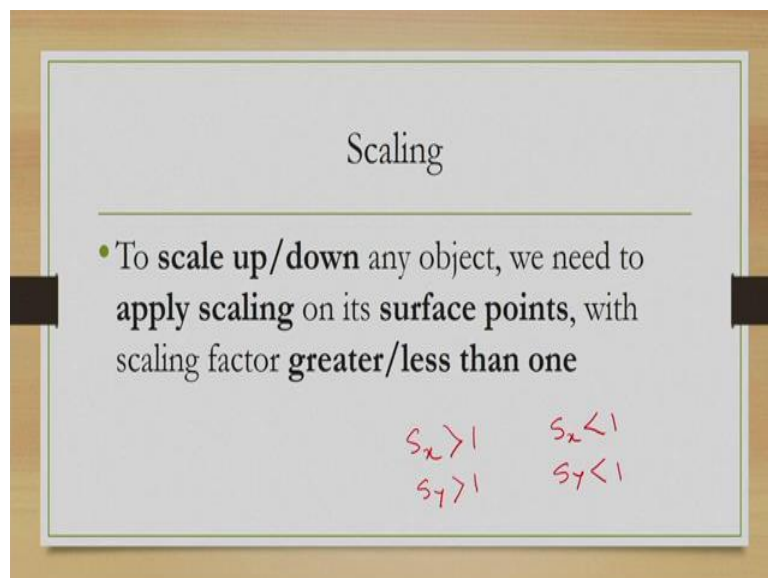
So, like in case of translation or rotation, here also, we did the same thing. That is we applied the scaling operations to all the points that define the object. Now, if we are using mesh representation, then those points are essentially the vertices of the vertex list. If we are using a spline representation, then these points are essentially the set of control points. And we apply scaling factor to each of these points to get the new objects, the new points that define the object.

(Refer Slide Time: 26:08)



Here, we should note one thing. So, if we are using the same scaling factor along both x and y direction, then this type of scaling is called uniform scaling. Otherwise, what we do is differential scaling. In the example, we have seen that the scaling factor along x direction is one third and along y direction is half so they are different, so we actually followed differential scaling in the example.

(Refer Slide Time: 26:41)

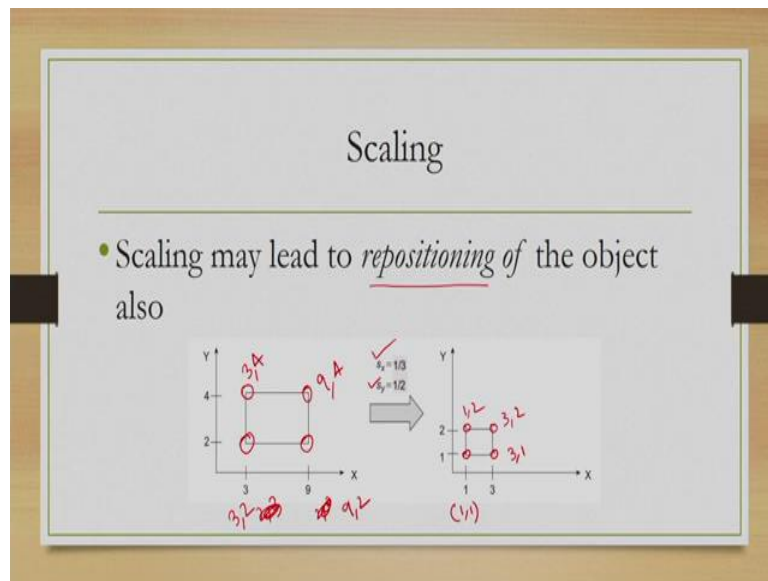


Now, when the scaling factor is, when say s_x is greater than 1 then along the x direction we are scaling up, when s_y is greater than 1, then along the y direction we are scaling up or increasing the size. Now, when both are greater than 1, then along both the directions we are

increasing the size. Similarly, when s_x is less than 1, we are reducing or scaling down the size along x direction.

Similarly, when s_y is less than 1, we are reducing or scaling down the size along y direction, when both are less than 1 then we are scaling down along both directions simultaneously. And of course, if s_x equal to 1 or s_y equal to 1, then there is no change in size.

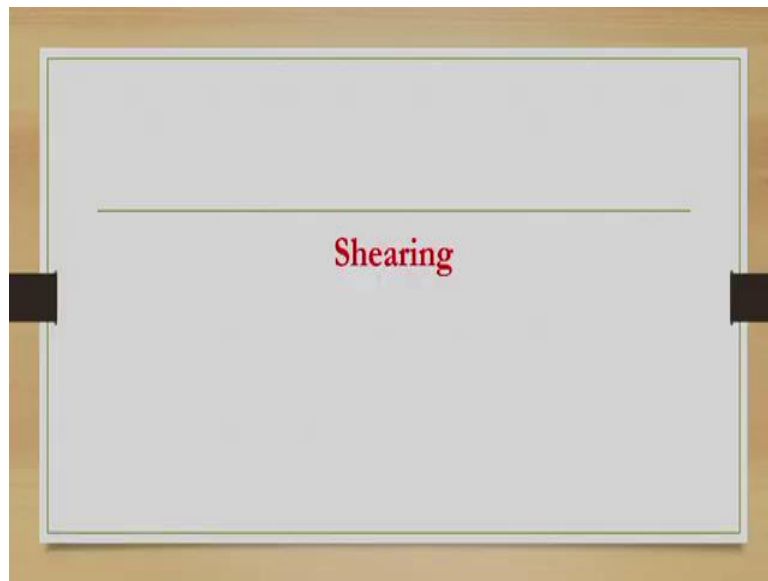
(Refer Slide Time: 27:41)



One important point to be noted here is that, during scaling the object may get repositioned, as we have seen in the example. So original vertex was at (3, 2) here it was at (9, 2) here it was at (9, 4) and it was at (3, 4). Now after scaling, by applying the scaling factors along x and y directions, we got a new object defined by the 4 vertices. What are the coordinates? We have (1, 1) then here we have (3, 1) here we have (3, 2) and here we have (1, 2). Now, as you can see the vertices got repositioned. So that is one of the effects of scaling.

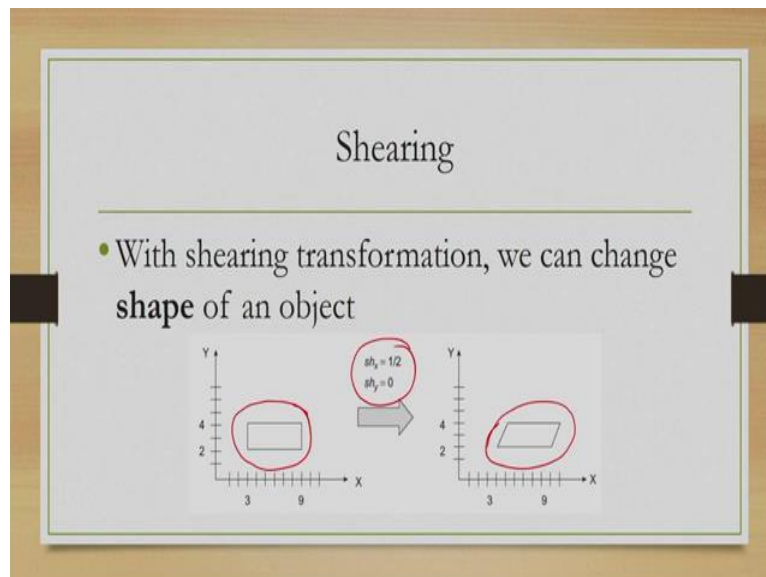
One effect is changing the size, the other effect is it may lead to repositioning of the objects. The final basic transformation is shearing.

(Refer Slide Time: 29:12)



What happens in shearing? Here we basically change the shape of the object. So far the transformations that we have learned deal with changing the position, orientation and size. Now, the final and the fourth basic transformation shearing allows us to change the shape also.

(Refer Slide Time: 29:40)



As you can see in this example, so we have one object here, which after shearing gets transformed to this object with a change in shape. Now, like scaling, shearing also essentially refers to multiplication of shearing factors along the x and y directions on the original object or on the original point. So if original point is x, then we multiply it with the shearing factor to get the transformed point x'. And same is true for y also.

(Refer Slide Time: 30:37)

Shearing

- General form of the transformation to determine the new point (x', y') from the current point (x, y)
$$\begin{aligned}x' &= x + sh_x y \\ y' &= y + sh_y x\end{aligned}$$
- sh_x and sh_y are **shearing factors** along X and Y directions

But, the relationship is slightly more complicated than scaling. Here, the new point is obtained by addition plus multiplication. So the new point is an addition of the old coordinate and a term which is a multiplication of the old coordinate with the shearing factor along that axis. But note here that to get x' , the new x coordinate, we use old x coordinate and also old y coordinate and the shearing factor along the x direction.

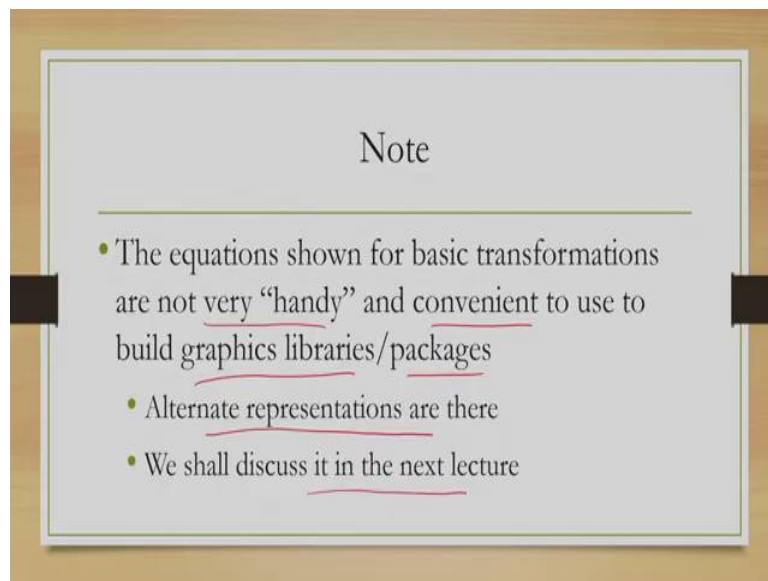
Similarly, to get new y coordinate we use old y coordinate and also old x coordinate and the shearing factor along the y direction. So that is the difference, slightly more complicated than scaling. And it allows us to change the shape of the object.

(Refer Slide Time: 31:57)

Shearing

- Similar to the scaling, we can apply shearing to all surface points to shear it

(Refer Slide Time: 35:39)

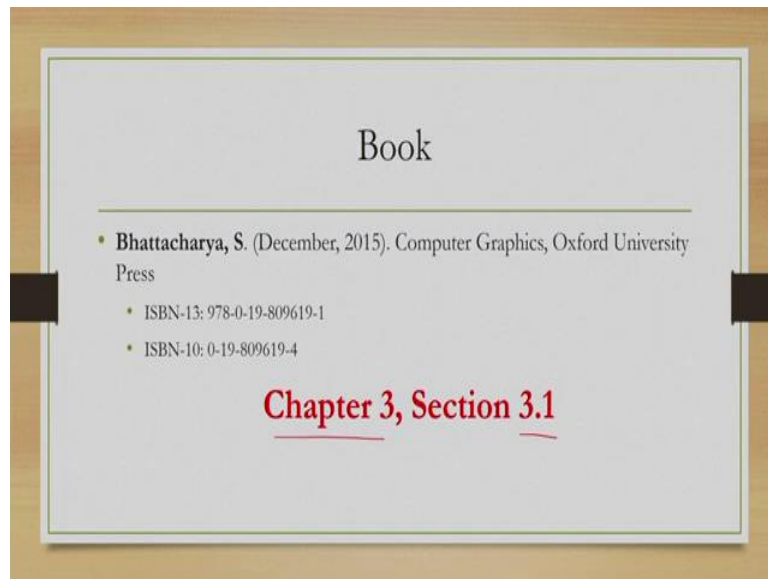


So, these are the four basic transformations. And when we are actually trying to transform an object we can apply these basic transformations in sequence, in multiple numbers, in different ways to get the desired transformation. Now, one point should be noted here that in order to perform the transformation, we have derived some equations, the equations that show the relationship between the old and the new point. So if we want to perform the transformation, we have to apply these equations on the old points to get the new points.

In fact, these equations are not very handy and convenient for use to build graphics libraries or packages. If we are trying to design a modular graphics system then these transformations represented in the form of equations may not be a good way to represent the transformations. We require alternative representations and those representations are there in the form of matrices. And we will also have many advantages if we are using matrices, particularly in the context of building modular systems.

So, in the next lecture, we shall discuss the matrix representation and why it is useful to represent transformations.

(Refer Slide Time: 37:28)



Whatever I have discussed today can be found in the chapter 3, section 3.1 of this book computer graphics. So, we will meet again in the next lecture. Till then, goodbye.