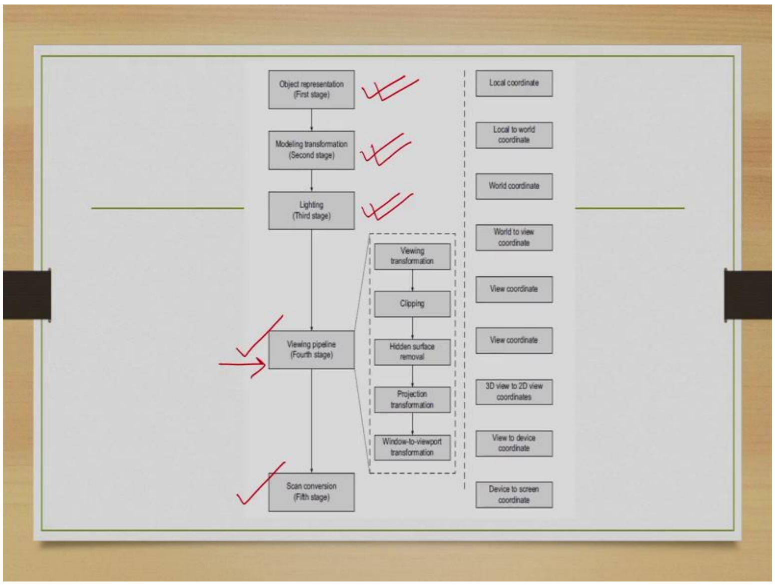
Computer Graphics
Professor Dr Samit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati
Lecture 19
Projection Transformation

Hello and welcome, to lecture number 19 in the course Computer Graphics. We are, continuing our discussion on the 3-D Graphics Pipeline. To recollect, the graphics pipeline is the set of stages that are used to convert a 3-D scene description to a 2-D image on the computer screen. And there are five stages in the pipeline; what are those five stages?

(Refer Slide Time: 1:00)



Object representation (First stage)

Modeling transformation (Second stage)

Lighting (Third stage)

Viewing pipeline (Fourth stage)

Viewing transformation

Clipping

Hidden surface removal

Projection transformation

Window-to-viewport transformation

Scan conversion (Fifth stage)

Local coordinate

Local to world coordinate

World coordinate

World to view coordinate

View coordinate

View coordinate

3D view to 2D view coordinates

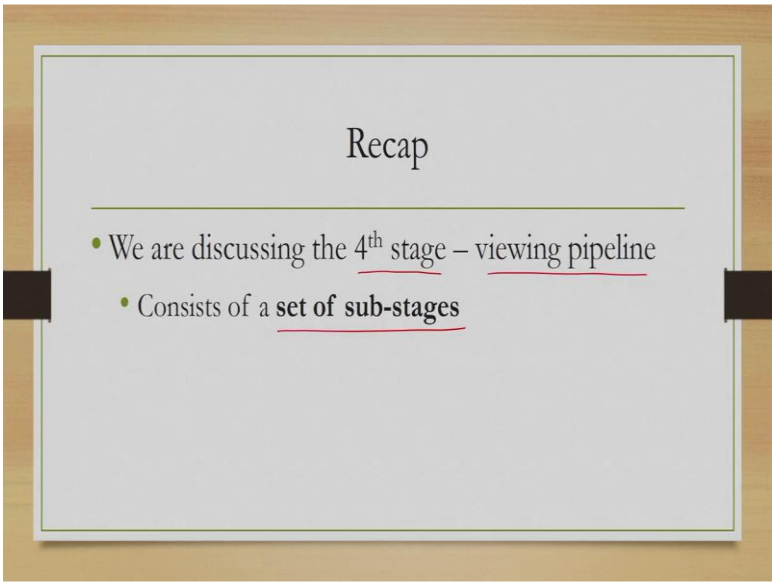
View to device coordinate

Device to screen coordinate

We have Object Representation as the first stage, Modeling Transformation as the second stage, Lightning or Assigning to the color objects as the third stage, Viewing Pipeline as the fourth stage, and Scan Conversation as the fifth stage.

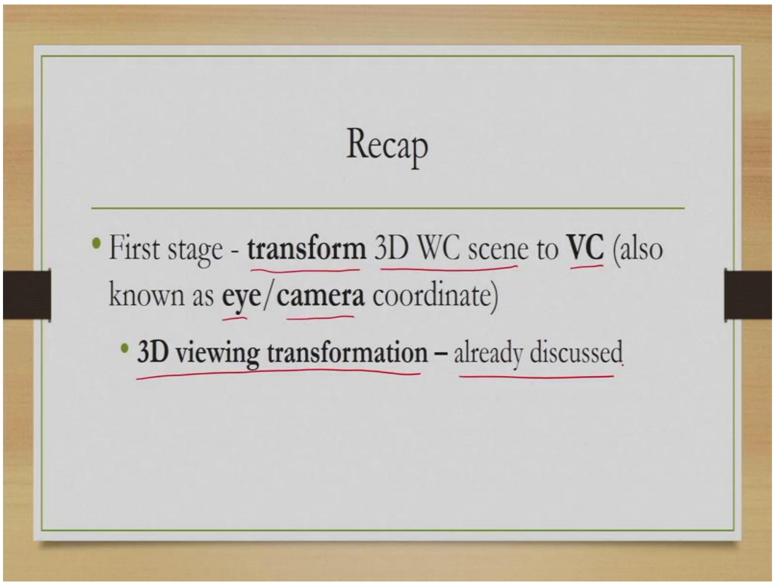
So, among them, we have already discussed the first three stages, object representation, modeling transformation, and lighting. Currently, we are discussing the fourth stage, that is viewing pipeline.

(Refer Slide Time: 1:42)



Recap

- We are discussing the 4th stage – viewing pipeline
  - Consists of a **set of sub-stages**



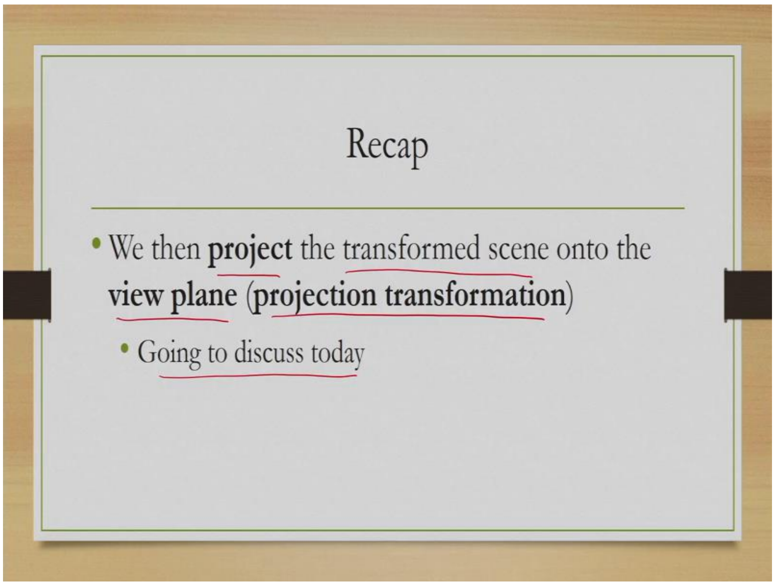
Recap

- First stage - **transform** 3D WC scene to **VC** (also known as **eye**/**camera** coordinate)
  - **3D viewing transformation –** already discussed

Now, as you can recollect this stage, the fourth stage that is the viewing pipeline stage consists of a set of sub-stages, what are those sub-stages? Now, the first sub-stage is a Transformation from a 3-D world coordinate scene to a view coordinate description.

Now, this view coordinate is also known as Eye or Camera coordinate system, and we have already discussed this transformation, which is called 3-D viewing transformation, in the previous lectures so, this is already discussed.

(Refer Slide Time: 2:32)

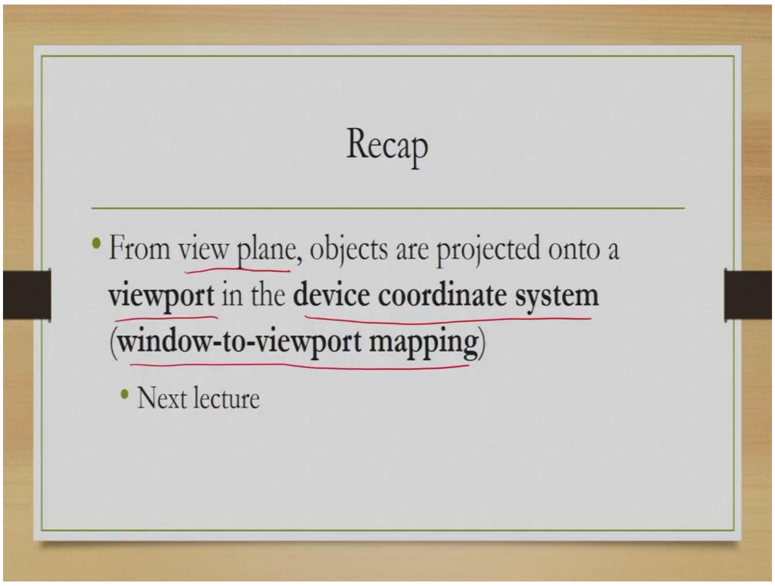


Recap

- We then **project** the transformed scene onto the **view plane** (**projection transformation**)
  - Going to discuss today

Next comes, Projection that is after the viewing transformation, we project the transformed scene onto the view plane, which is our projection transformation, and this transformation we are going to discuss today.

(Refer Slide Time: 2:58)

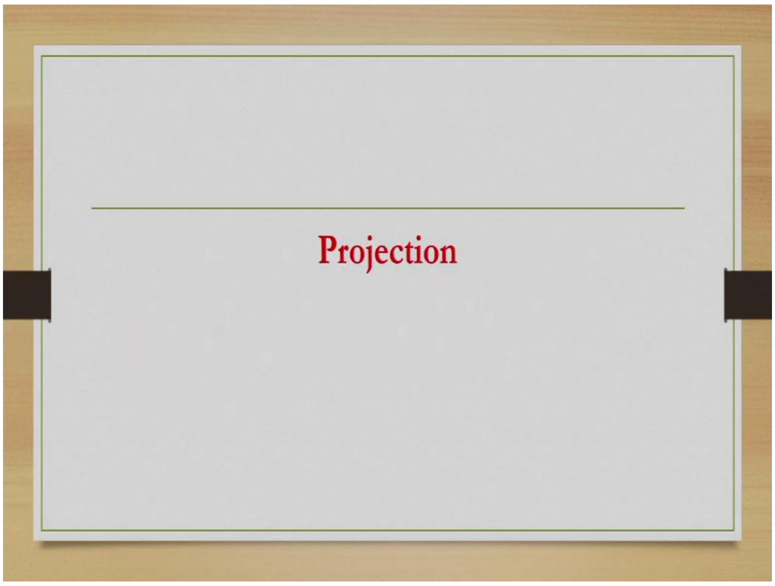


Recap

- From view plane, objects are projected onto a **viewport** in the **device coordinate system** (**window-to-viewport mapping**)
  - Next lecture

There is one more sub-stage, the third sub-stage, in which we perform another transformation. So, from the view plane where the scene is projected, we transform it to a description on the device coordinate system in a region called a viewport. This is called window to viewport mapping, where window refers to a region on the view plane and this will be our next lecture subject matter, today we are going to discuss the second stage that is projection.

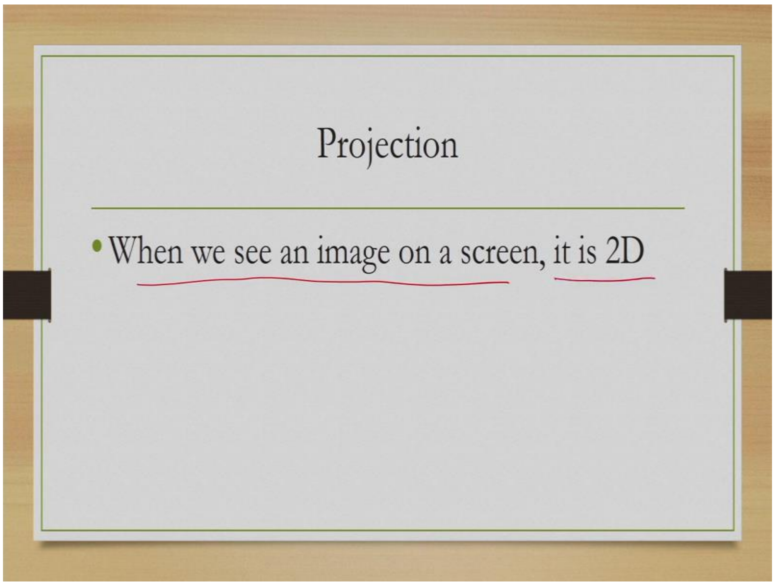
(Refer Slide Time: 3:47)



Projection

Let us try to understand the basic idea behind projection before we discuss the projection transformation.

(Refer Slide Time: 3:58)

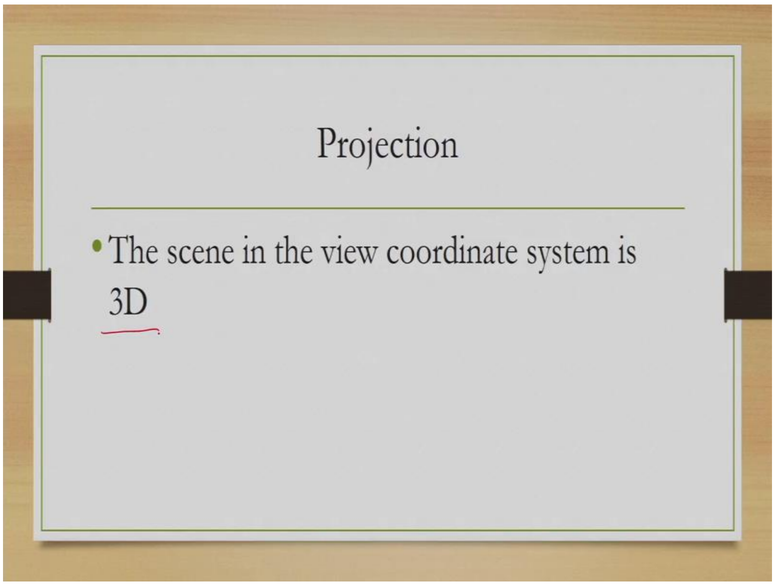


Projection

- When we see an image on a screen, it is 2D

So, why we require projection? We all know when we see an image on a screen, it is a 2-D image; the image is display on a 2-D computer screen.

(Refer Slide Time: 4:14)

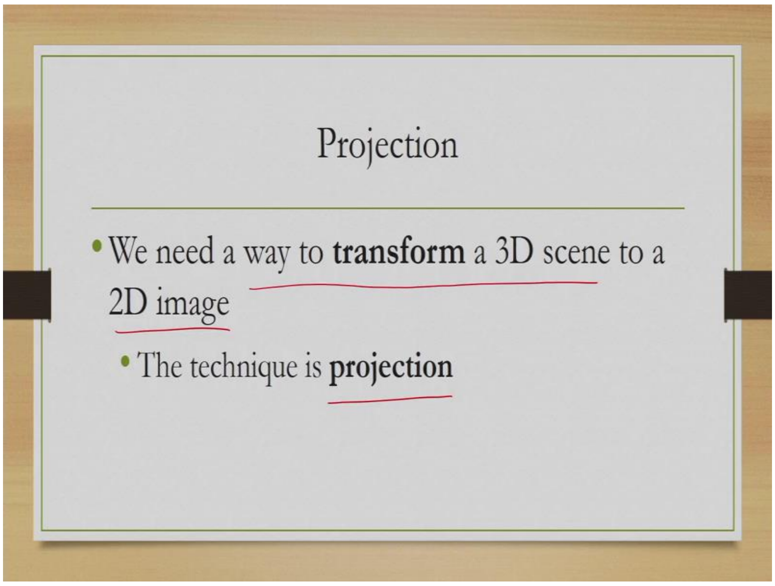


Projection

- The scene in the view coordinate system is 3D

However, when we discussed about transforming the world coordinate scene to a view coordinate scene, that was still a 3-D description, so the scene that was transformed to view coordinate system was still in 3-D or three-dimensional scene.

(Refer Slide Time: 4:43)



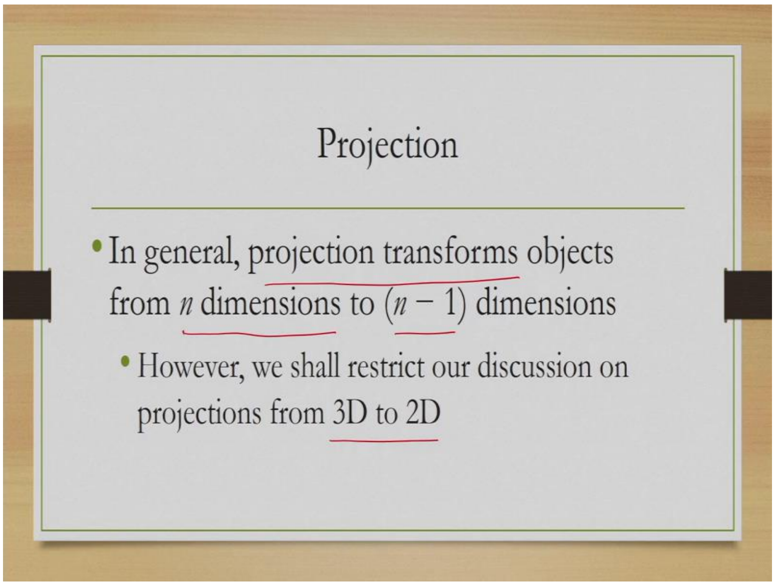
Projection

- We need a way to **transform** a 3D scene to a 2D image
  - The technique is **projection**

Then what is required? We need a way to transform a 3-D scene to a 2-D image, and this technique where we are transforming a 3-D description to a 2-D image description is called projection.

So, the idea is simple, when we see something on a screen that is on a 2-D screen, however, our definitions as well as representation are in 3-D, and we require some way to transfer from 3-D description to 2-D description, and that is projection.

(Refer Slide Time: 5:22)

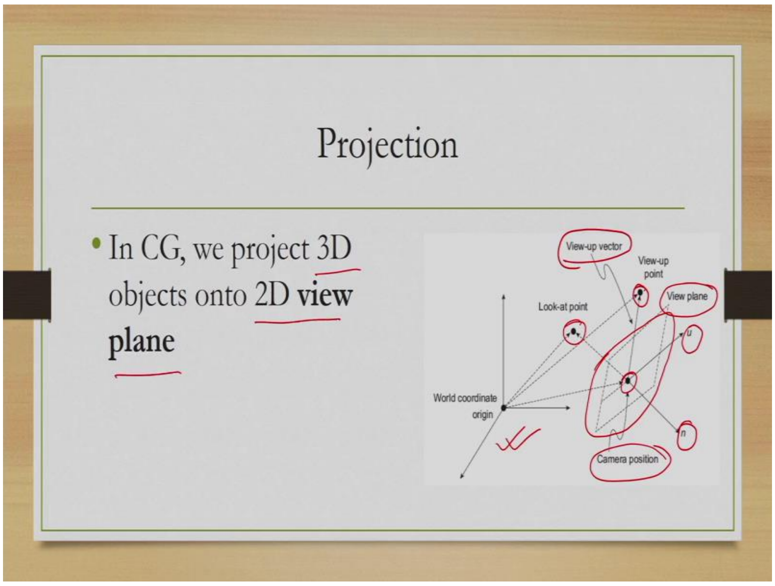


Projection

- In general, projection transforms objects from n dimensions to $(n-1)$ dimensions
  - However, we shall restrict our discussion on projections from 3D to 2D

In general, projection transforms objects from n dimension to (n – 1) dimension so, it reduces the dimension by 1 in our case of course, we will not go into the general description or projection, instead we will restrict our discussion to projection from 3-D to 2-D which will serve the purpose of this course.

(Refer Slide Time: 5:51)



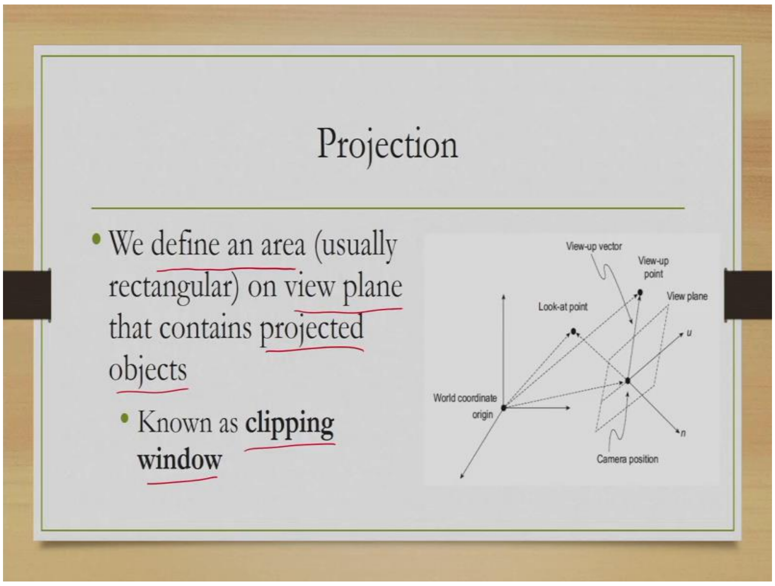
Projection

- In CG, we project 3D objects onto 2D **view plane**

So, let us try to understand the basic setting. So, we have this world coordinate system where the scene is described. Now, we are looking at the scene through the viewing mechanism provided in the camera, and then we are taking a snapshot. So, this is the look at point and this snapshot when we take the snapshot on the film or on a screen that is called a view plane, this is the camera position, this is the view up point all these concepts we already discussed in the previous lecture and along with that the view up vector also.

So, this view coordinate system is defined by these three principle axis n, u, and v. So, essentially, what we are doing? We are projecting the 3-D objects onto the 2-D view plane.

(Refer Slide Time: 7:24)



Projection

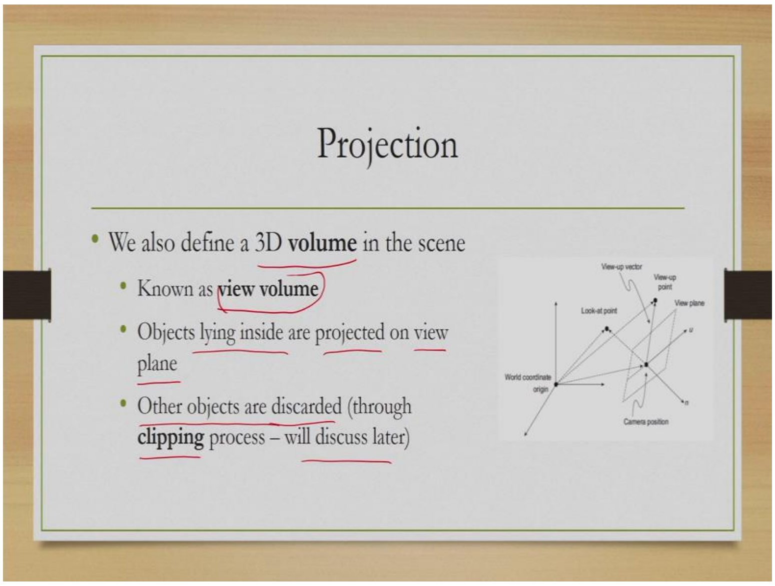
- We define an area (usually rectangular) on view plane that contains projected objects
  - Known as **clipping window**

View-up vector
View-up point
View plane
Look-at point
u
World coordinate origin
n
Camera position

But, the entire view plane is not utilized, we define an area on this plane that contains the projected objects, and this area is typically called clipping window.

So, in graphics, we will assume that whatever we want to project, we are projecting on a particular region on a view plane that is called the clipping window, later on, we will see why it is called clipping window.

(Refer Slide Time: 7:57)



Projection

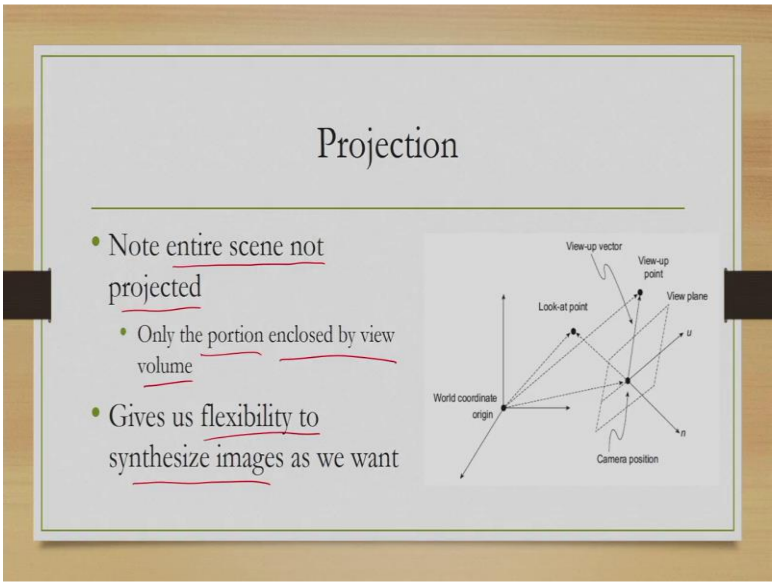
- We also define a 3D **volume** in the scene
  - Known as **view volume**
  - Objects lying inside are projected on view plane
  - Other objects are discarded (through **clipping** process – will discuss later)

View-up vector
View-up point
View plane
Look-at point
u
World coordinate origin
n
Camera position

There is a third component also, so we also define a 3-D volume or a region in space in the scene. So, there is a scene, and within that scene, we are defining a 3-D volume. This is called the view volume. Now, this view volume is actually our way of specifying which objects we need to project on the clipping window. So, whichever objects are lying inside this view volume are projected on the view plane, more specifically on the clipping window and other objects that are outside are discarded.

Now, this discarding takes place through a process called clipping, which we will discuss in the next lectures in subsequent lectures. So, essentially we have a view plane. Within this we define a clipping window, and also in the 3-D scene description, we define a view volume. Whatever objects are inside this view volume are projected onto this clipping window on the view plane, and whatever lies outside the volume are discarded through a process called clipping.

(Refer Slide Time: 9:27)



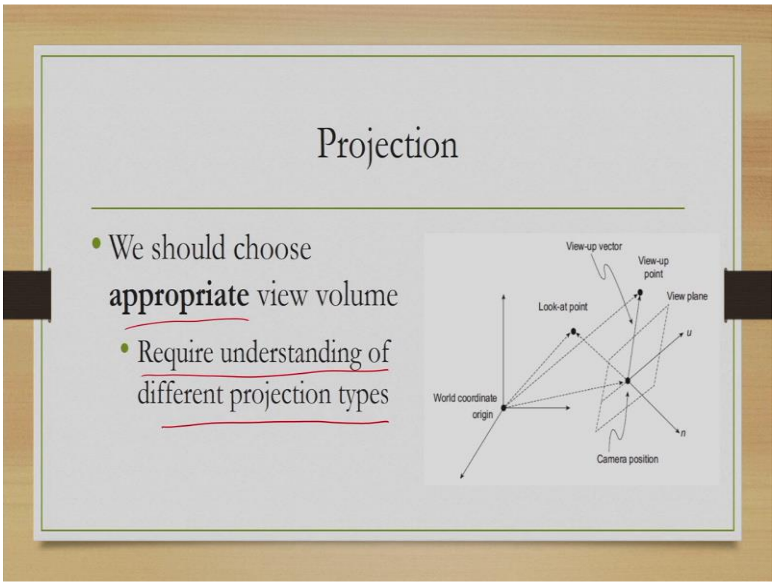
Projection

- Note entire scene not projected
  - Only the portion enclosed by view volume
- Gives us flexibility to synthesize images as we want

View-up vector
View-up point
View plane
Look-at point
u
World coordinate origin
n
Camera position

So, the point to be noted here is that the entire scene is not projected; instead only a portion enclosed the view volume is projected. So, in this way, by controlling the view volume, we can control the amount of scene that we want to display on the screen. So, this gives us some flexibility in synthesizing the images; by increasing the volume, we want to show a larger image; by reducing the volume, we can show smaller images.

(Refer Slide Time: 10:12)



Projection

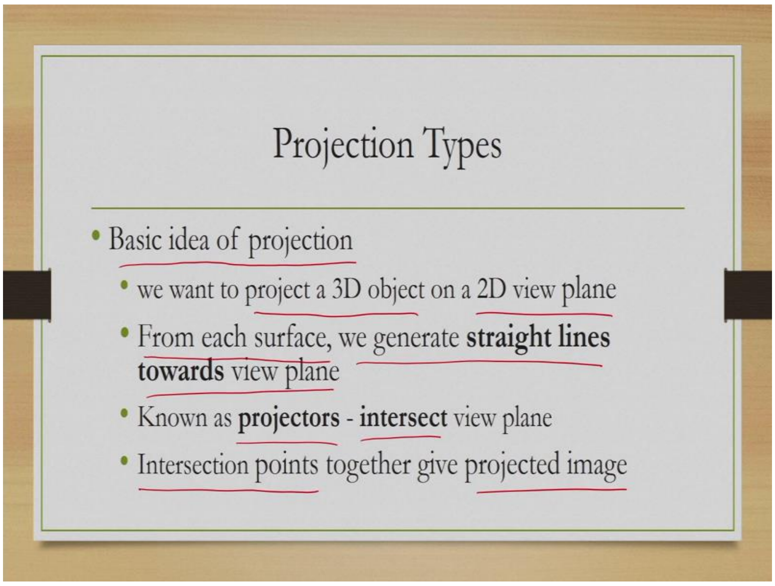
- We should choose **appropriate** view volume
  - Require understanding of different projection types

View-up vector
View-up point
View plane
Look-at point
u
World coordinate origin
n
Camera position

It maybe also noted that this larger and smaller in terms of size in terms of the amount of region of the 3-D scene description that we want to display.

So, by larger image, I meant that I can show a larger amount of the scene on the screen if we increase the view volume size; similarly, if we reduce the view volume size, then we can show the smaller region of the scene on the screen or on display. So, this brings us to the point of how to choose appropriate view volume so that we can control the content of the image; this requires some understanding of the different projection types.

(Refer Slide Time: 11:23)



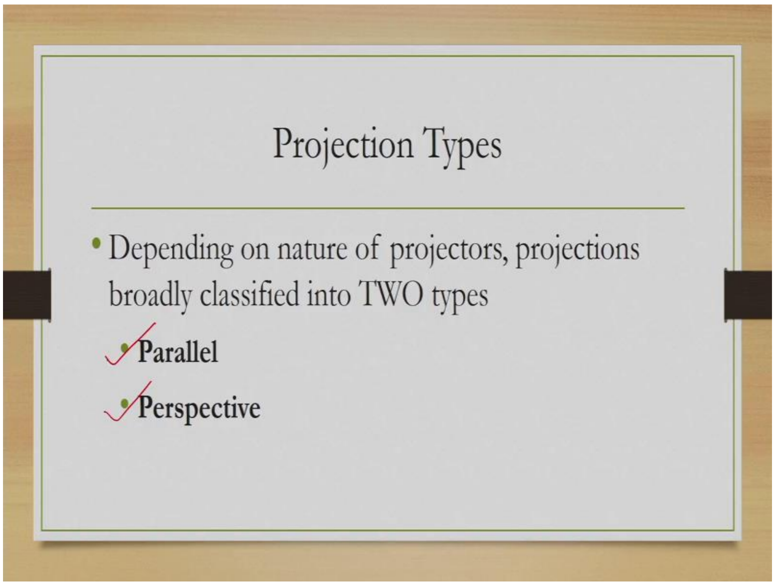
Projection Types

- Basic idea of projection
  - we want to project a 3D object on a 2D view plane
  - From each surface, we generate **straight lines towards** view plane
  - Known as **projectors** - **intersect** view plane
  - Intersection points together give projected image

So, let us try to understand different projection types. In order to do that, we have to now, to go the basic idea of projection at another level we have to understand the idea. So, what we want? We want to project a 3-D object on a 2-D view plane, so from each surface point of the objects that are present in the 3-D scene, we generate straight lines towards the view plane in order to create that projected image.

Now, these straight lines are known as projectors, and they intersect the view plane. When we put together these intersection points, we get the projected image. So, essentially how we generate the projection? We generates projectors that are straight lines originating from the surface points in the original scene, and these lines goes towards the view plane intersects them and these intersection points taken together give us the projected image.

(Refer Slide Time: 12:49)

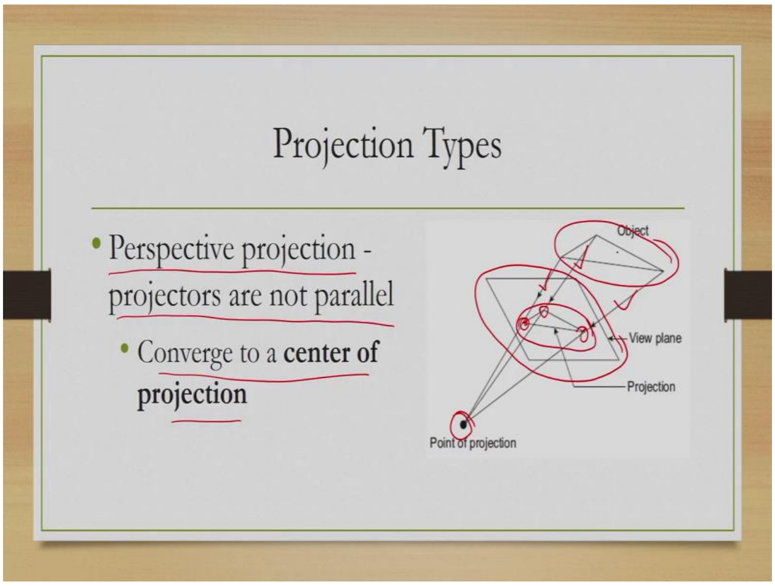


Projection Types

- Depending on nature of projectors, projections broadly classified into TWO types
  - Parallel
  - Perspective

Now, depending on the nature of projectors, we can have two broad types of projections one is Parallel projection, one is Perspective projection. There are many sub-types also, but, here we will not going to the finite details of different types and sub-types; instead, we will restrict our discussion to these broader types.

(Refer Slide Time: 13:20)



Projection Types

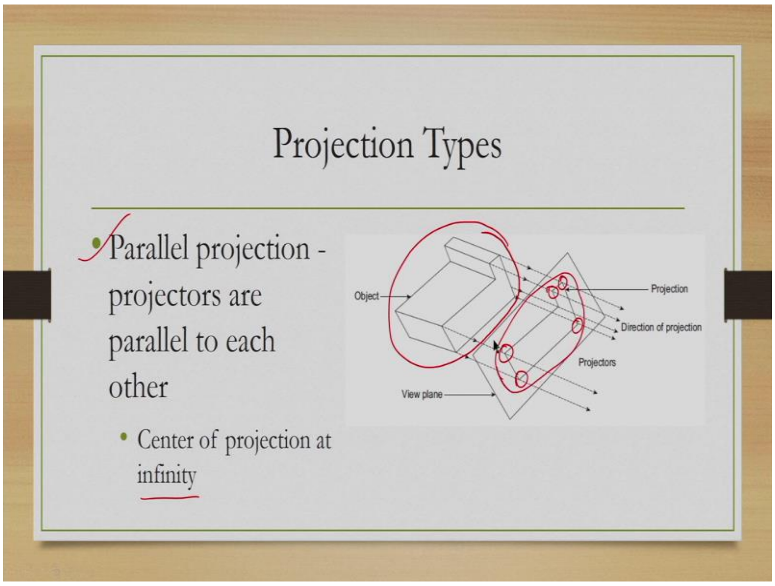
- Perspective projection - projectors are not parallel
  - Converge to a **center of projection**

So, in the case of perspective projection, what happens? The projectors are not parallel to each other, and they converge to a center of projection

So, here as you can see, this is the object, this one is the view plane, now from the objects we created projectors like here, here, here. Now, these projectors are not parallel to each other, and they were projected towards a common point of projection where they meet during this process they intersect or passes through the view plane and the points of intersection like here, here, here it came together gives us the projected image.

So, this object here is projected in this way, and this type of projection where the projectors are not parallel to each other instead, they meet at a point of projection is called perspective projection.

(Refer Slide Time: 14:31)



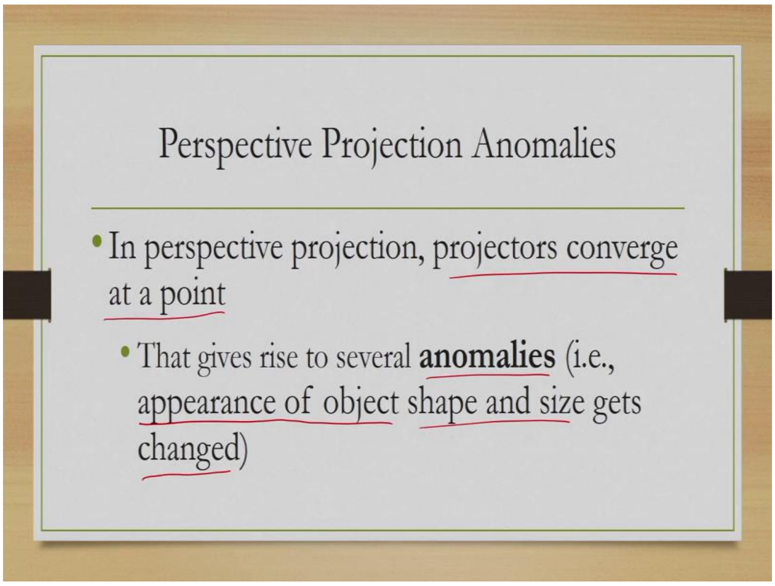
Projection Types

- Parallel projection - projectors are parallel to each other
  - Center of projection at infinity

Now, there is another type of projection that is the parallel projection. In this case, projectors are parallel to each other they do not meet at a common point; instead, typically, it is assumed that they meet at infinity. That means, they do not meet in simple terms and this projectors when they intersect the view plane they generates sets of intersection points these points are taken together gives us the projected image as shown in this figure.

So, here in this figure, as you can this is the object this entire thing, and we get this projection on the view plane.

(Refer Slide Time: 15:39)



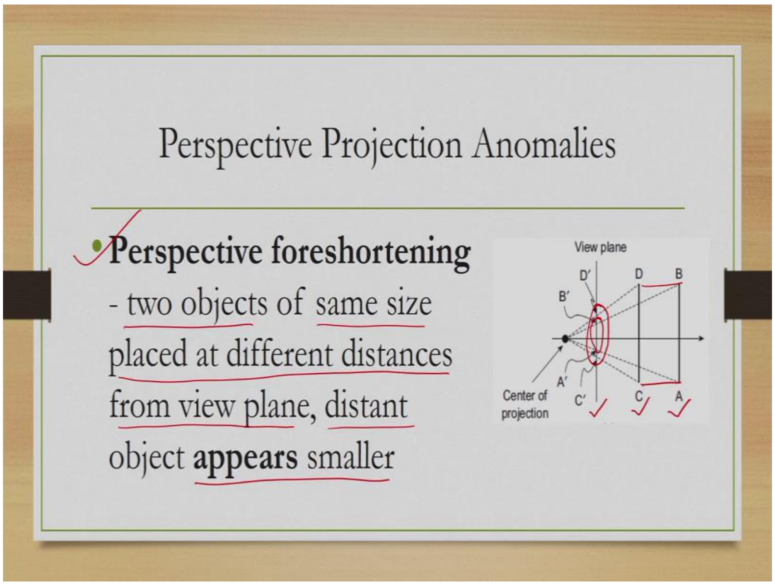
Perspective Projection Anomalies

- In perspective projection, projectors converge at a point
  - That gives rise to several **anomalies** (i.e., appearance of object shape and size gets changed)

Now, in the case of perspective projection, certain things happen, and we should be aware of those things in fact because, of those things, only we get the only perception of reality let us see what are those things, they are together known as Anomalies.

Now, those anomalies happen because the projectors converge at a point, and these anomalies indicate that the appearance of the object in terms of shape and size gets changed in perspective projection.

(Refer Slide Time: 16:06)



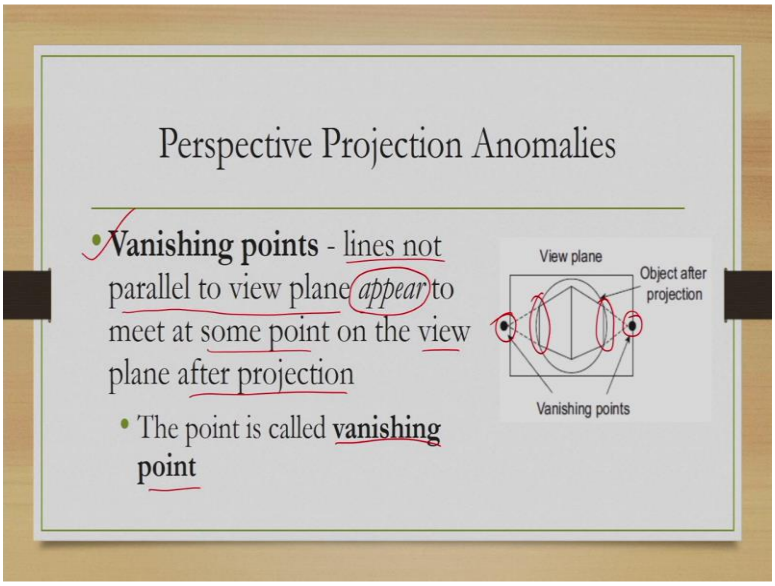
Perspective Projection Anomalies

- **Perspective foreshortening** - two objects of same size placed at different distances from view plane, distant object **appears** smaller

Now, one anomalies perspective foreshortening. So two objects of the same size placed at different distances from the view plane; if that is the case, then the distant object appears smaller.

So, if this is the view plane, this is one object, this is another object then as you can see object A, B size appears to this which is smaller than object C, D, although they are actually of same size so, this happens because of the projectors meeting at a common center of projection and as you can see form here that because of this reason, we get a sense of reality that distant objects appear smaller.

(Refer Slide Time: 17:09)



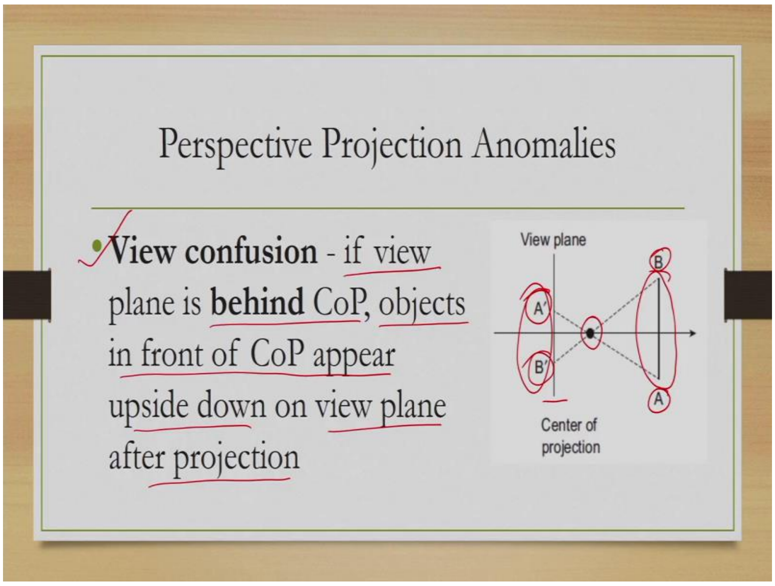
Perspective Projection Anomalies

- **Vanishing points** - lines not parallel to view plane appear to meet at some point on the view plane after projection
  - The point is called **vanishing point**

Another anomaly is called vanishing points; here, the lines that are not parallel to the view plane appear to be meeting at some point on the view plane after projection. Now, the point where these lines appear to meet is called the vanishing points. For example, here, if we have an object like this as you can see this side and this side, they appear to meet at these points; these are vanishing points.

So, accordingly, the object shape is projected; this again gives us a sense of 3-D reality due to the occurrence of the vanishing points.

(Refer Slide Time: 18:15)



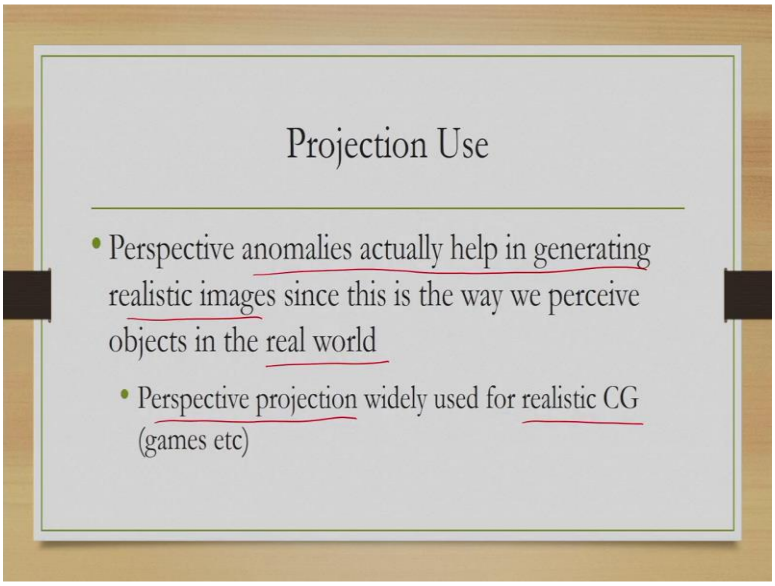
Perspective Projection Anomalies

- **View confusion** - if view plane is **behind** CoP, objects in front of CoP appear upside down on view plane after projection

There is another anomaly called view confusion. Now, if the view plane is behind the center of projection that means, the points where the projectors meet then the objects that are in front of the center of projection appear upside down on the view plane after projection, and this is simple to understand, this is center of projection and the view plane is behind it so, this object will appear upside down as shown here, this point will be projected to this point and this point will be projected to this point, this is view confusion.

So, we have perspective foreshortening where distant objects appear smaller then vanishing points and then view confusion. So, together these anomalies make us perceive an object with changed shape and size, which intern reinforces our perception of 3-D reality. So, how we can use the projections?

(Refer Slide Time: 19:42)



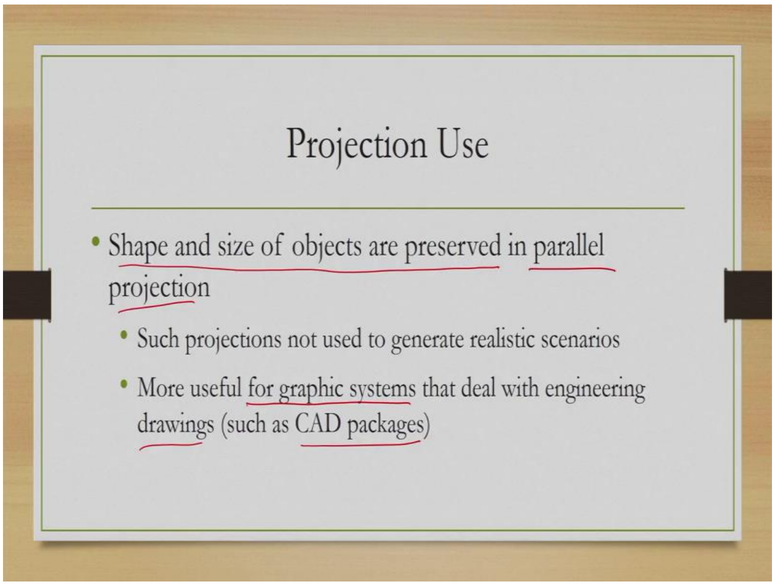
Projection Use

- Perspective anomalies actually help in generating realistic images since this is the way we perceive objects in the real world
  - Perspective projection widely used for realistic CG (games etc)

As I said, perspective anomalies actually help in generating realistic images since this is the way how we perceive objects in the real world.

So, perspective projection can be used for realistic computer graphics; for example, in games or animations, we can use perspective projection wherever we require 3-D realistic scenes to be generated; we should use perspective projection.

(Refer Slide Time: 20:24)



Projection Use

- Shape and size of objects are preserved in parallel projection
  - Such projections not used to generate realistic scenarios
  - More useful for graphic systems that deal with engineering drawings (such as CAD packages)

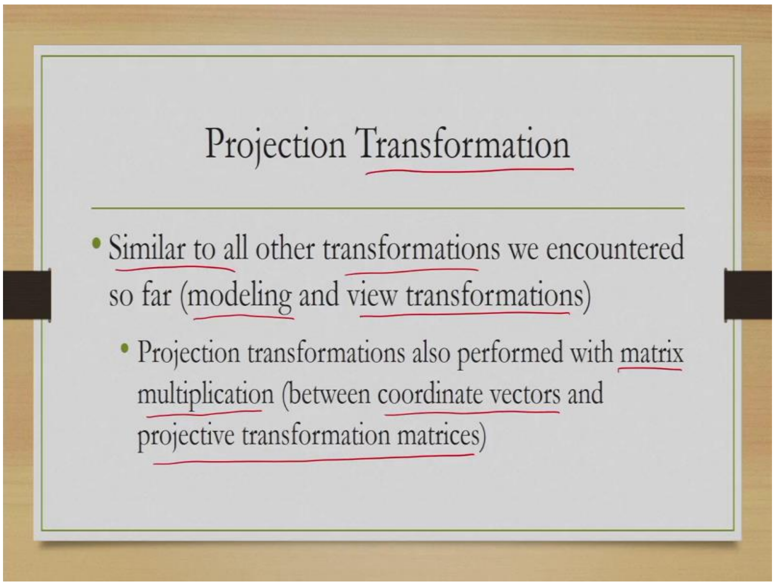
On the other hand, in the case of parallel projection, the shape and size of the objects are preserved. They do not change, unlike in the case of perspective projection. As a result, the parallel projection is not suitable for giving us realistic images.

So, if we use parallel projection to generate a scene that would not look like realistic. So, it is not used for realistic 3-D, instead where it is used? For graphics systems that typically deal with Engineering drawings such as the CAD packages that we have discussed at the beginning of the course or compute aided design packages, so there realism is not important instead, other things are more important, so there parallel projection may be useful.

So, with that, I hope you got some basic idea of projections, so to recap, we use projections to map 3-D scene to a 2-D image on the view plane, and we do this on a clipping window, which is a region on the view plane that is based on the view volume that we define in the 3-D space, and then we perform projection transformation.

So, let us now shift our focus to the idea of projection transformation what it is and how we can do this.

(Refer Slide Time: 22:17)



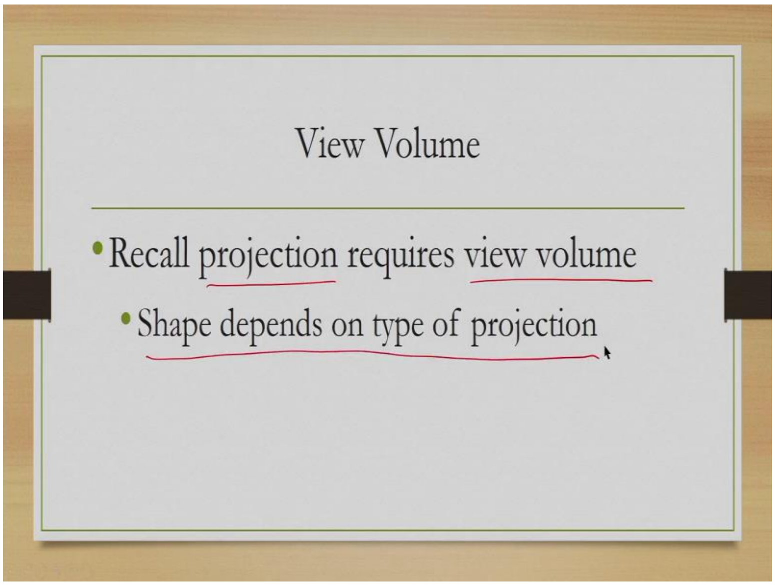
Projection Transformation

- Similar to all other transformations we encountered so far (modeling and view transformations)
  - Projection transformations also performed with matrix multiplication (between coordinate vectors and projective transformation matrices)

As the name suggests, it is a transformation. So, it is similar to all other transformations that we have already encountered, namely, the modeling transformation and the view transformation in which way it is similar, so we can perform these transformations with matrix multiplication between the point vectors and the projective transformation matrices.

So, essentially our objective is to have all sorts of transformations represented as a matrix multiplication, and we have already seen how to do it with modeling transformation; we have seen how to do it with view transformation. Now, let us try to derive projective transformation matrices so that we can do it with projection transformation as well.

(Refer Slide Time: 23:21)

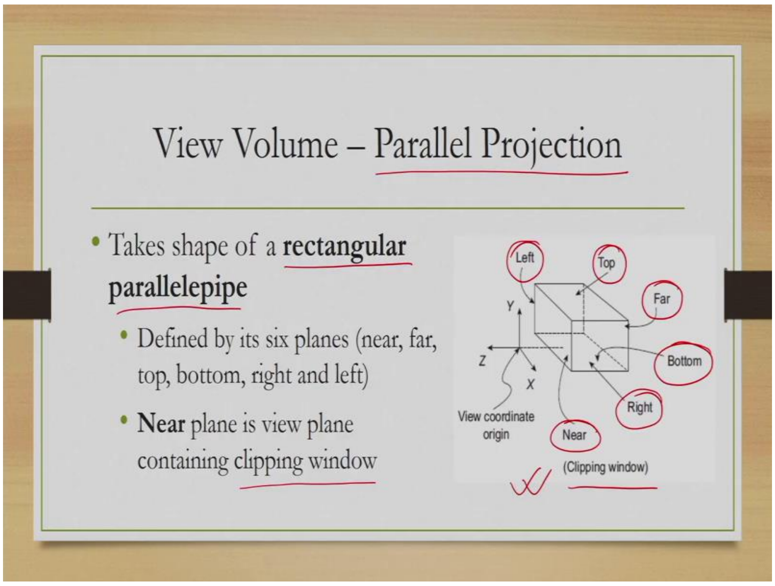


View Volume

- Recall projection requires view volume
  - Shape depends on type of projection

Now, in order to understand those matrices, we require some understanding of the view volume because the projections are dependent on the view volumes, and the shape of the view volumes actually depends on the type of projection we are interested in, so we already mentioned there are two types one is the perspective projection, one is the parallel projection and their corresponding view volumes are different.

(Refer Slide Time: 23:52)



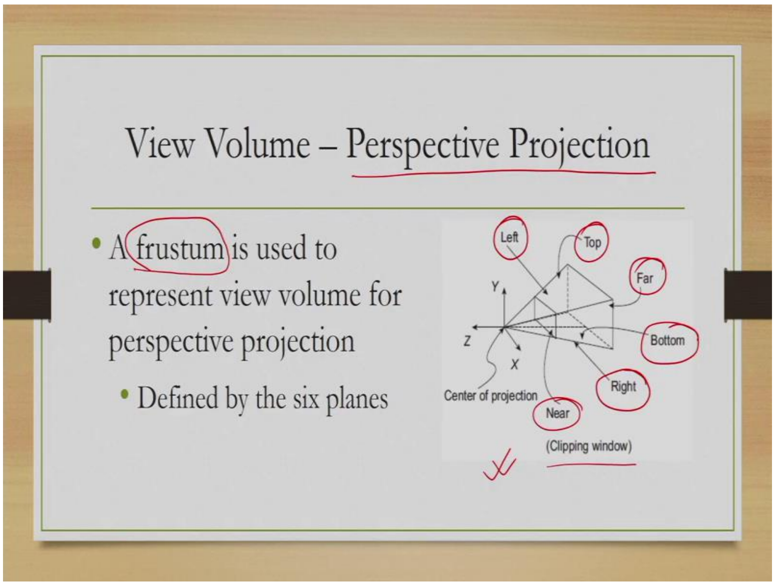
View Volume – Parallel Projection

- Takes shape of a **rectangular parallelepipe**
  - Defined by its six planes (near, far, top, bottom, right and left)
  - **Near** plane is view plane containing clipping window

Now, in the case of parallel projection the view volume takes the shape of a rectangular parallelepiped as shown in this figure. Here, there are six phases marked as near to the view plan, then right, the bottom, then far plane, top and left plane. Now this, near plane is also the clipping window.

So, essentially it is the view plane containing the clipping window. So, in the case of parallel projection we define view volume as a rectangular parallelepipe defined by six planes, and the near plane is the view plane containing the clipping window.

(Refer Slide Time: 25:00)

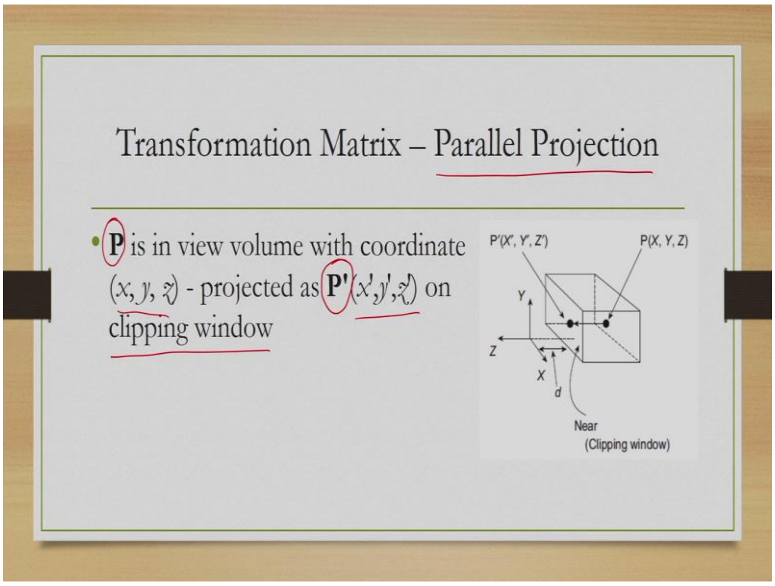


View Volume – Perspective Projection

- A frustum is used to represent view volume for perspective projection
  - Defined by the six planes

What happens in the case of perspective projection? So, in the case of perspective projection what we use is a frustum as shown in this figure like the parallel projection here also the frustum is defined in terms of its bounding planes, so we have near plane, far plane, right plane, left plane, top plane and bottom plane and the near plane contains the clipping window. So, essentially the idea is the same with parallel projection in the sense that in the both cases the near plane is the plane where the clipping window is situated.

(Refer Slide Time: 26:02)

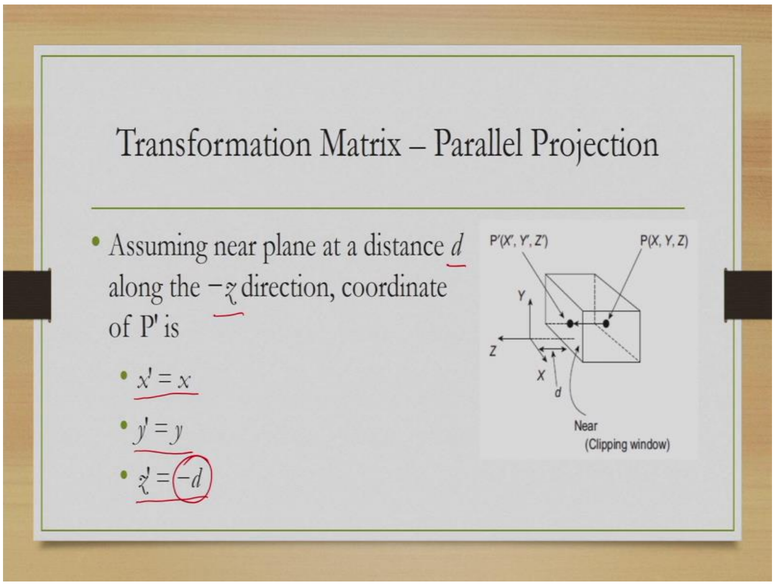


Transformation Matrix – Parallel Projection

- P is in view volume with coordinate (x, y, z) - projected as P'(x', y', z') on clipping window

So, with this knowledge of the view volume, let us try to understand the projection transformation matrices for the two types of projection. Let us start with the parallel projection. So, in this case, let us consider, a point P which is in the view volume with coordinates x, y, z. Now, we want to project this point on the view plane as a projected point P’ with the new coordinates x’, y’, z’. And this projection takes place on the clipping window, as we have already mentioned, and our objective is to relate these two points.

(Refer Slide Time: 26:53)

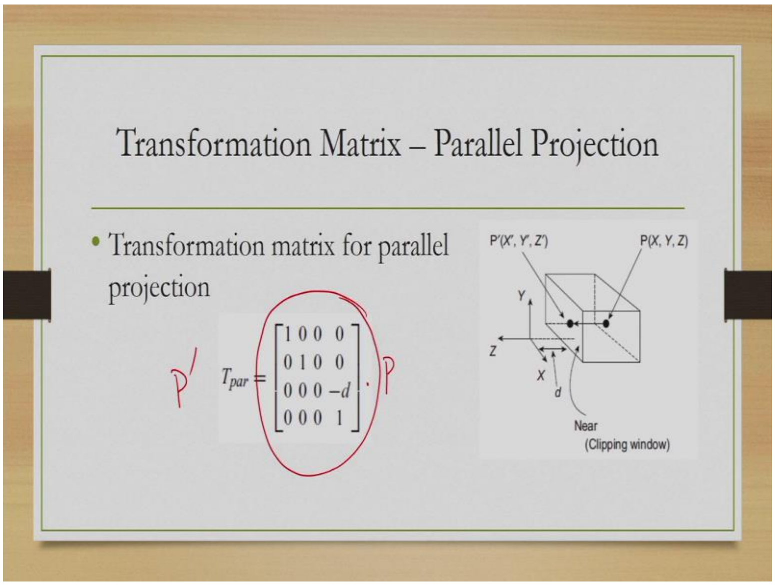


Transformation Matrix – Parallel Projection

- Assuming near plane at a distance d along the $-z$ direction, coordinate of P' is
  - $x' = x$
  - $y' = y$
  - $z' = \boxed{-d}$

Let us assume that the near plane is at a distance d along the -z direction, so then we can simply understand it is quite obvious that, since it is a parallel projection the new x coordinate will be same as the original one, the y coordinate will also be the same in case of z coordinate there will be change, so the new z coordinate will be - d.

(Refer Slide Time: 27:29)



Transformation Matrix – Parallel Projection

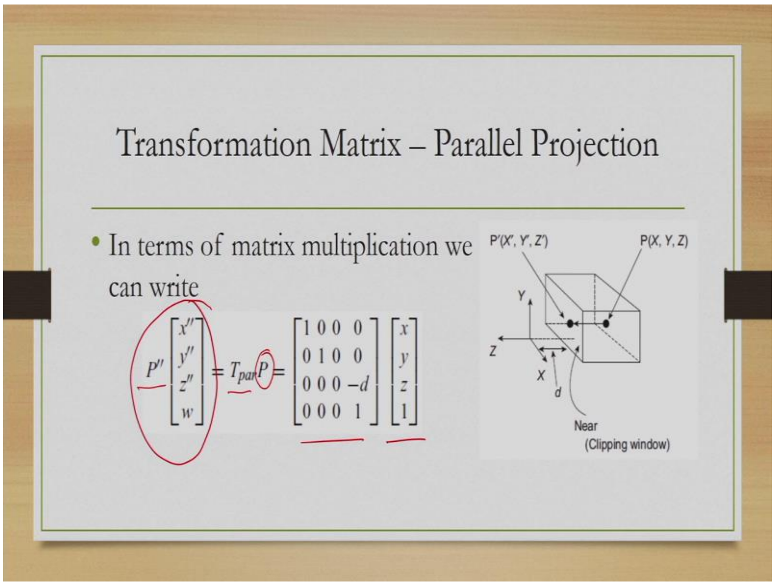
- Transformation matrix for parallel projection

$$P' \quad T_{par} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot P$$

Then, we can actually represent this information in the form of a transformation matrix for parallel projection, as shown here.

So, when we multiply this transformation matrix with the point vector P, then we will get the new point vector P’.

(Refer Slide Time: 27:59)



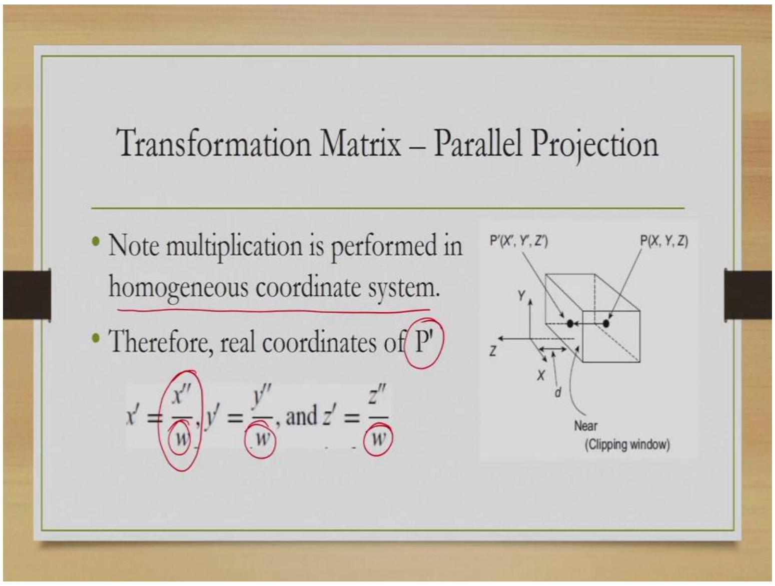
Transformation Matrix – Parallel Projection

- In terms of matrix multiplication we can write

$$P'' \begin{bmatrix} x'' \\ y'' \\ z'' \\ w \end{bmatrix} = T_{par}P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

However, we have to keep in mind that this multiplication that we are performing that is we are trying to get the transform point by multiplying the transformation matrix with the original point in this way; this transformation takes place in a homogeneous coordinate system.

(Refer Slide Time: 28:27)



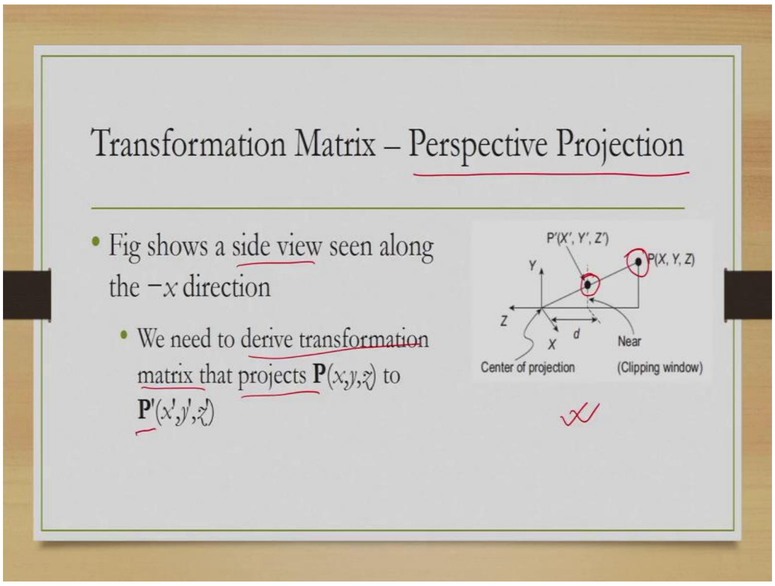
Transformation Matrix – Parallel Projection

- Note multiplication is performed in homogeneous coordinate system.
- Therefore, real coordinates of P'

$$x' = \frac{x''}{w}, y' = \frac{y''}{w}, \text{and } z' = \frac{z''}{w}$$

So, the multiplication performed in this homogeneous coordinate system and the real coordinates of P’ should be obtained by dividing it with the homogenous factor w. Now, we shall see later that in the case of transformations, w need not be 1, so we require division. We will see some examples later where w is not 1, unlike in the previous transformations where the homogeneous factor was 1.

(Refer Slide Time: 29:03)



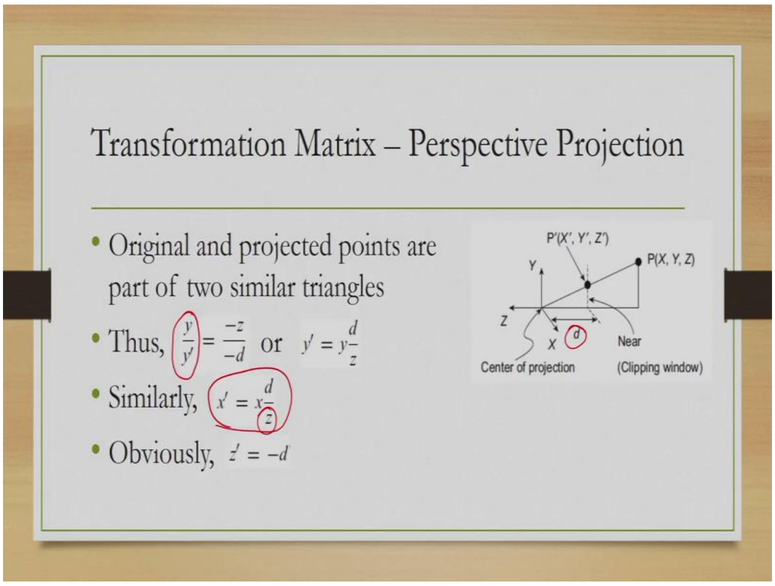
Transformation Matrix – Perspective Projection

- Fig shows a side view seen along the $-x$ direction
  - We need to derive transformation matrix that projects $\mathbf{P}(x,y,z)$ to $\mathbf{P}'(x',y',z')$

Now, let us see the perspective projection. So, this is more complex than parallel projection; in parallel projection we simply drop or change the coordinate but, here we require to change all the coordinates because the projectors meet at a point. Now, to derive this changes, let us try to understand the projection with this figure; the figure shows the side view along with the - x direction.

So, what we need? We need to derive the transformation matrix that projects P. This is the point P to the point on the view plane or clipping window P’.

(Refer Slide Time: 30:05)

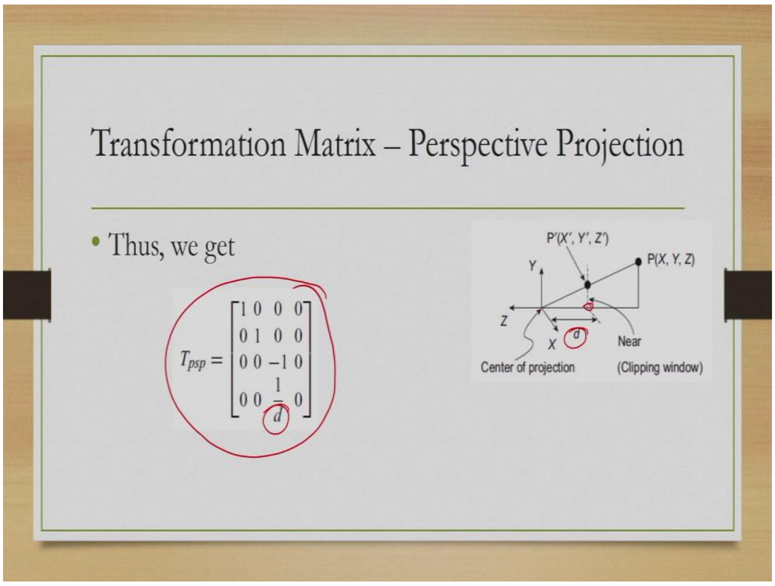


Transformation Matrix – Perspective Projection

- Original and projected points are part of two similar triangles
- Thus, $\frac{y}{y'} = \frac{-z}{-d}$ or $y' = y\frac{d}{z}$
- Similarly, $x' = x\frac{d}{z}$
- Obviously, $z' = -d$

Now, from the original and projected points, you can see that they are part of two similar triangles, and from these triangles, we can form some relationships. So, like between the y coordinates and between the x coordinates in terms of d, that is the distance of the view plane or the near plane from the origin and the original z coordinate value.

(Refer Slide Time: 30:52)



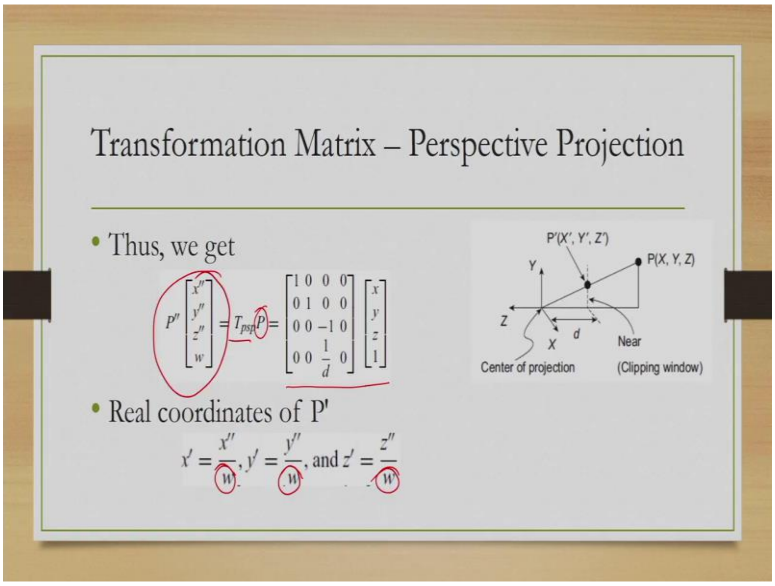
Transformation Matrix – Perspective Projection

- Thus, we get

$$T_{psp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}$$

From there, we can reorganize to get the transformation matrix in this form, where d is the distance shown here between the origin and the projected point on the near plane or between the origin and the near plane.

(Refer Slide Time: 31:17)



Transformation Matrix – Perspective Projection

- Thus, we get

$$P''\begin{bmatrix} x'' \\ y'' \\ z'' \\ w \end{bmatrix} = T_{psp}P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Real coordinates of P'

$$x' = \frac{x''}{w}, y' = \frac{y''}{w}, \text{and } z' = \frac{z''}{w}$$

Now, like in the case of parallel projection here also, what we can do? We can multiply this perspective projection matrix with the original point vector to get the projected point in the homogeneous coordinate system and to get back to the original point, what we require is we need to divide it by this homogeneous factor w, and we will see that here again, w will not be equal to 1.

So, we require some divisions, unlike in the cases of other transformations that we have seen before. So, that is how we can derive the projection matrices. Now, few things we should note here, one is the derivations that are shown are basically very simplified situations taken into account; in reality that actual projection matrices are slightly more complicated, plus there is some other information that is also stored along with projection, those things we will discuss later briefly although, we will not going to the minute details of those concepts.

(Refer Slide Time: 32:51)

Summary

So, to summarize today, what we have learned is the idea of projection, why we require projection? To transform from 3-D scene description to description on a 2-D plane, which we are calling the view plane, on the view plane, we define a region which is called clipping window on which this projection takes place, and for the purpose of projection we define a 3-D region in the 3-D space that is called view volume.

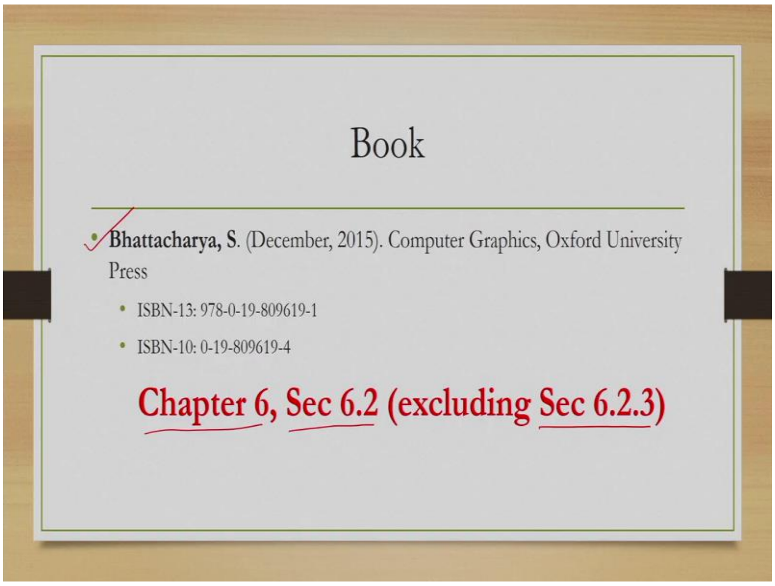
Now, there are two types of view volumes defined for two types of projections, a rectangular parallel pipe for parallel projection and a frustum for perspective projection. In case of perspective projection, we get to see several anomalies that changes the shape and size of the objects after projection, and that gives us the perception of 3-D reality. Accordingly, for applications of computer graphics where we require to generate 3-D realistic scenes, we use perspective projection, whereas parallel projection can be used in situations where 3-D realism is not required.

And we have also shown how to derive the projection transformation matrices for the basic two types of projections, namely parallel projection and perspective projection. The transformation idea is the same. Essentially, we have a transformation matrix this matrix we multiply with the point vector to get a new point vector, the transformed point vector; however, we have to keep in mind that this transform point vector is defined in the homogenous coordinate system.

So, to get to the actual transform point vector, we need to divide the obtained coordinate values with the homogeneous factor w, and in the case of transformations where projection is involved, w is not 1, unlike the other transformations that we have seen before namely modeling transformation and view transformation.

In the next lecture, we will talk about the other sub-stage of the view pipeline that is the window to viewport mapping.

(Refer Slide Time: 35:39)



Book

- Bhattacharya, S. (December, 2015). Computer Graphics, Oxford University Press
  - ISBN-13: 978-0-19-809619-1
  - ISBN-10: 0-19-809619-4

Chapter 6, Sec 6.2 (excluding Sec 6.2.3)

Whatever I have discussed today can be found in this book you are advised to refer to chapter 6, the whole section 6.2, excluding section 6.2.3. This section we will discuss in the next lecture. Now, in this section, you will find more details about projections and types of projections. You may go through those details if you are interested. That is all for today, thank you and goodbye.