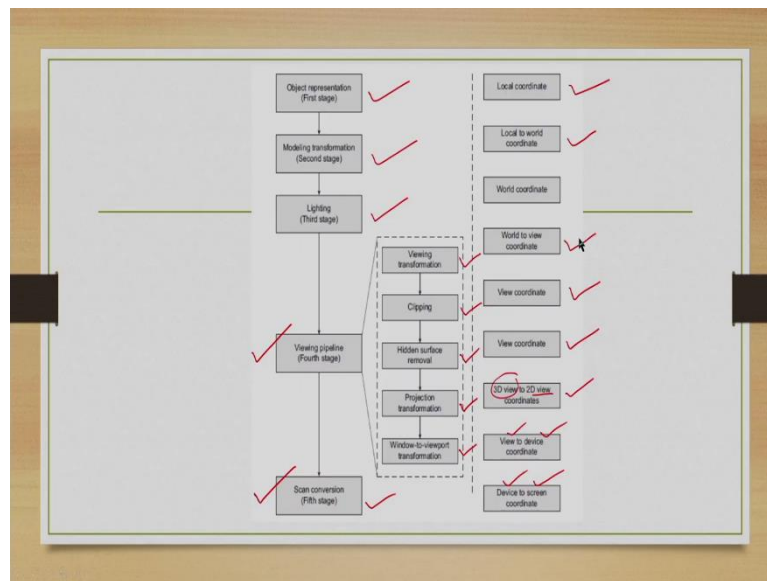


**Computer Graphics**  
**Professor Dr. Samit Bhattacharya**  
**Computer Science and Engineering**  
**Indian Institute of Technology, Guwahati**  
**Lecture 18**  
**View transformation**

Hello, and welcome to lecture number 18 in the course Computer Graphics. We are currently discussing the 3D Graphics pipeline. That is the set of stages that converts an object description to a 2D image on a computer screen. What are the stages? Let us quickly recap.

(Refer Slide Time: 00:51)



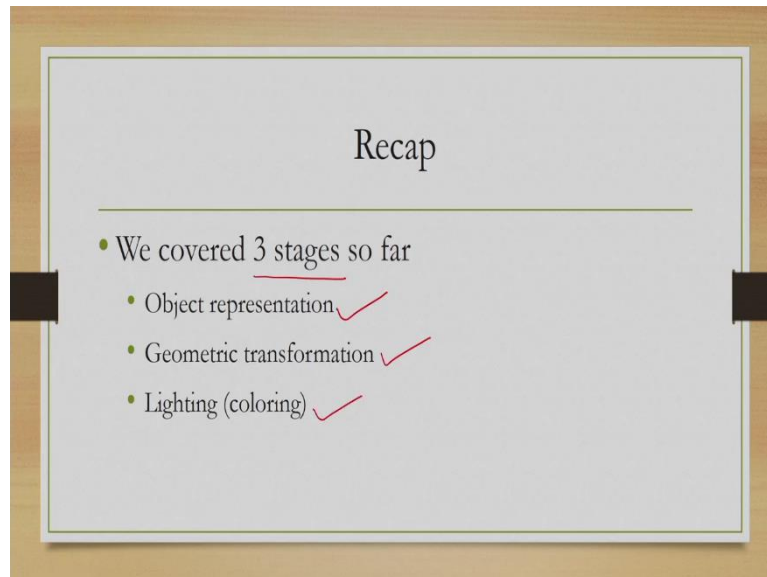
There are five stages. As shown here. Object representation, Modelling transformation, Lighting, Viewing pipeline, and Scan conversion. And also, it may be recalled that each of these stages works in a specific coordinate system. For example, object representation works in local or object coordinate system; modeling transformation works in local to world coordinate systems. It is basically a transformation from local coordinate to world coordinate system.

Then lighting or assigning colours to objects happen in the world coordinate system. Then the viewing pipeline, the fourth stage actually consists of five sub-stages, and there is a variation in coordinate system where they work. For example, the first stage, viewing transformation is a transformation from world to a view coordinate system, then clipping the second stage works in view coordinate system.

Hidden surface removal, third stage works in view coordinate system, projection transformation, which is again a transformation that takes place between 3D view coordinate to 2D view coordinate system and window to viewport transformation the fifth substage of

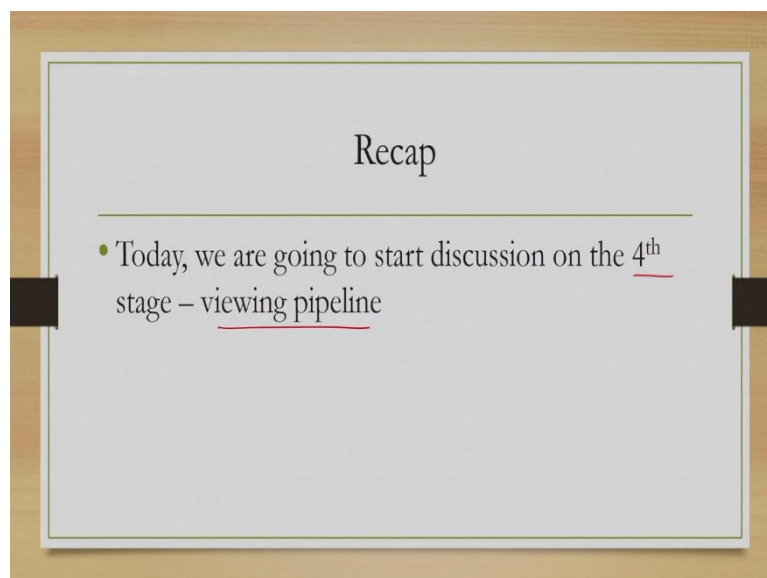
the fourth stage takes the object description from view coordinate to device coordinate system. The last stage, scan conversion, essentially is a transformation from device to screen or pixel coordinates.

(Refer Slide Time: 03:03)



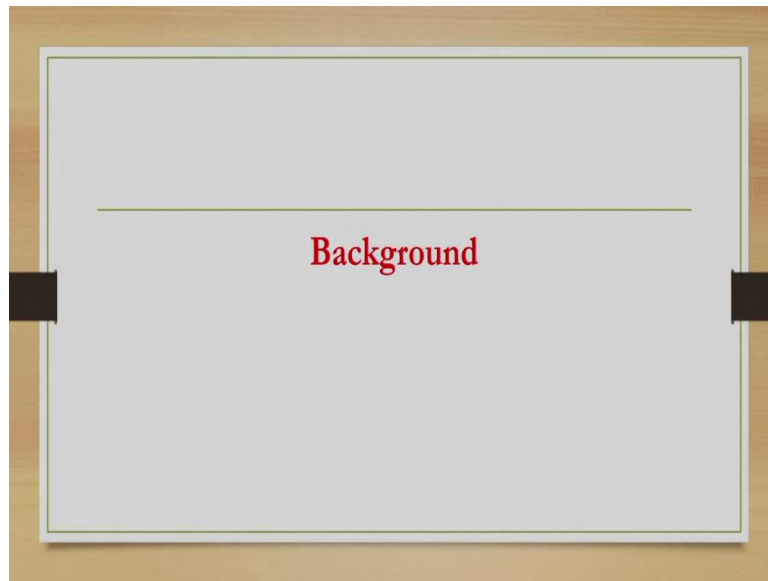
So, far among all these stages and sub-stages, we have covered three steps—the first three stages in our previous lectures. The first one is object representation, then geometric transformation and lighting or assigning colours to the objects.

(Refer Slide Time: 03:20)



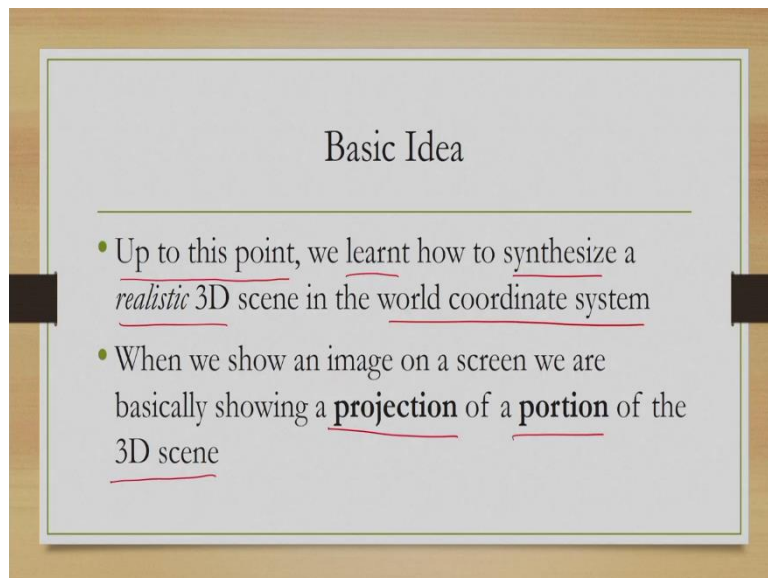
Today, we will start our discussion on the fourth stage that is the viewing pipeline. Now let us start with some basic idea of what we mean by the viewing pipeline.

(Refer Slide Time: 03:39)



Let us start with a background knowledge.

(Refer Slide Time: 03:44)

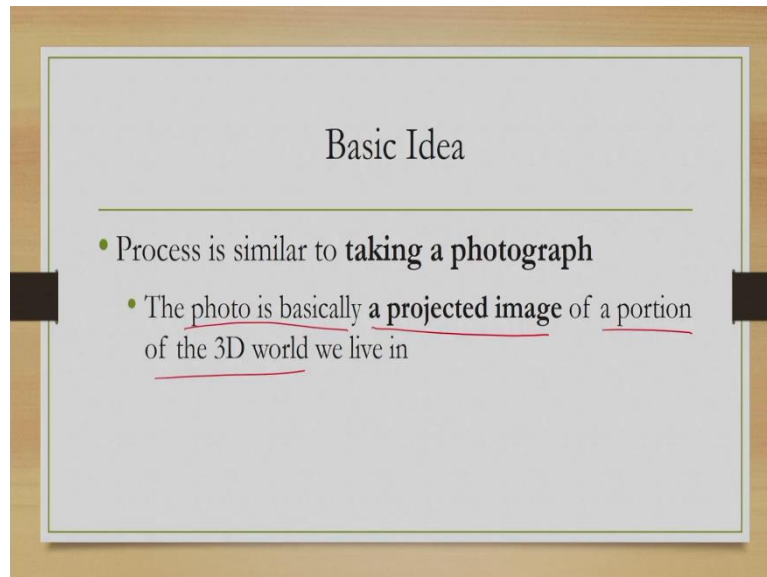


So, up to this point whatever we have discussed in our previous lectures. We learnt how to synthesize a realistic 3D scene in the world coordinate system. So, we started with the object definition stage or object representation stage. Then we put together the objects in the second stage modeling transformation stage to construct a world coordinate scene.

And then, in the third stage, we assigned colours. In the world coordinate description of the scene to make it look a 3D scene. So, the knowledge we have gained so far is good enough to explain how we can create a 3D realistic scene. Now, that is not the end of it. So, we need to display this scene on a 2D computer screen. So, that means essentially, we need a projection

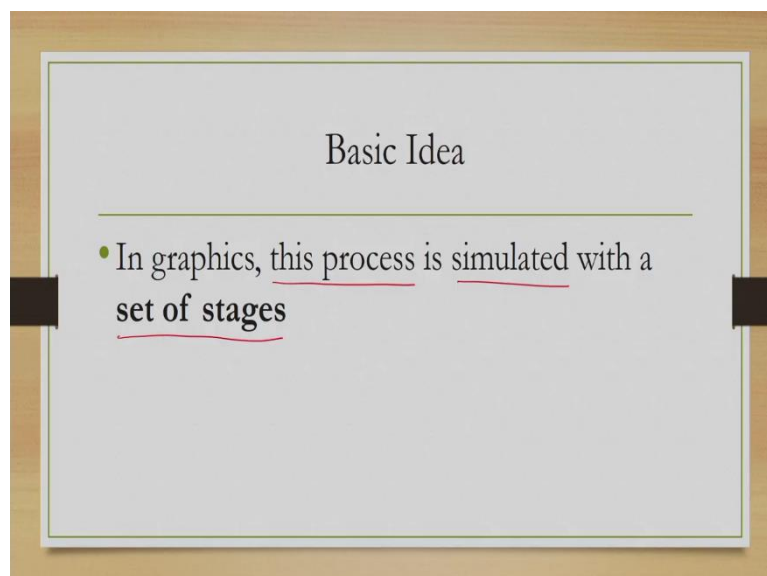
of the scene on a 2D screen from the 3D description, and this projection need not be of the whole scene. It can be a part of the scene also, which is the most common way of looking at it; we usually talk about a portion of the overall 3D description to be projected on a screen.

(Refer Slide Time: 05:33)



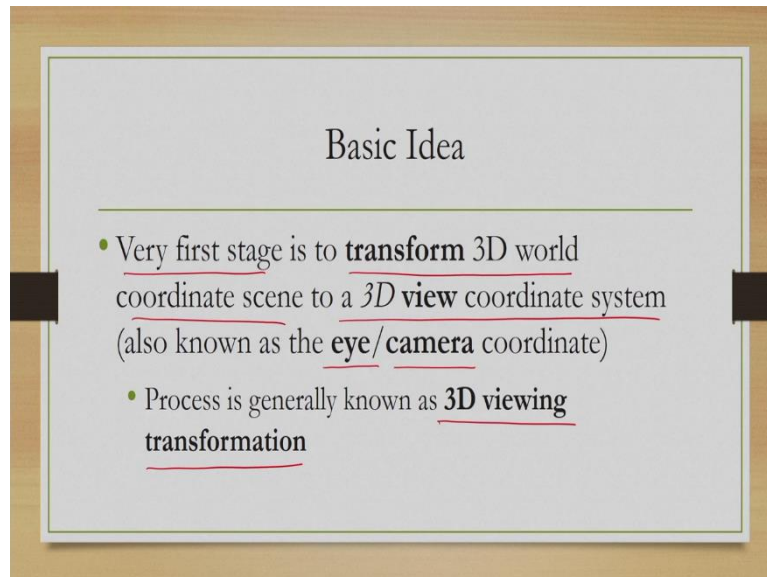
Now, this process of projection is actually similar to taking a photograph. So, when we take a photo, the photo is basically a projected image of a portion of the 3D world that we see around us that we live in, and this projected image is on the photographic plate or a camera screen. So, when you are talking of displaying a 2D image on a computer screen essentially, we start with a 3D description of the scene, and then we want to simulate the process of taking a photograph.

(Refer Slide Time: 06:12)



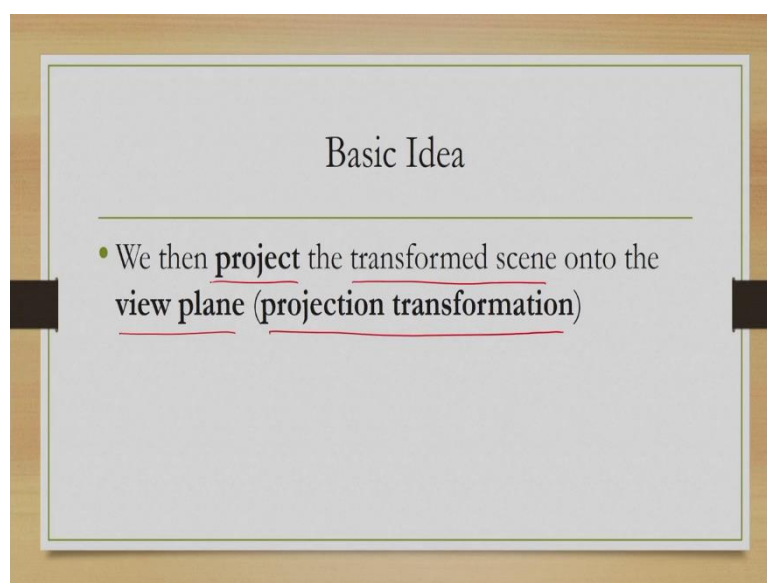
Now, this process, process of taking a photograph is simulated in computer graphics with a set of stages and these stages together constitute the fourth stage of the graphics pipeline that is viewing pipeline. What are those stages?

(Refer Slide Time: 06:32)



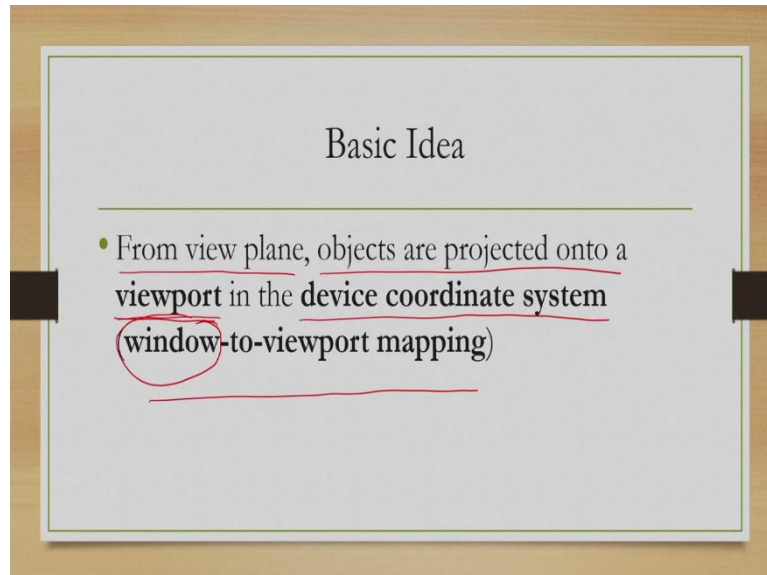
The very first page is to transform the 3D world coordinate scene to a 3D view coordinate system or reference frame. Now, this 3D view coordinate system is also known as the eye coordinate system or a camera coordinate system. And this process of transforming from 3D world coordinates to 3D view coordinate is generally called the 3D viewing transformation. So, this is the first stage in the 3D viewing pipeline.

(Refer Slide Time: 07:23)



Then we project the transformed scene onto the view plane. So, this is the projection transformation, so first comes 3D view transformation, then comes projection transformation.

(Refer Slide Time: 07:38)

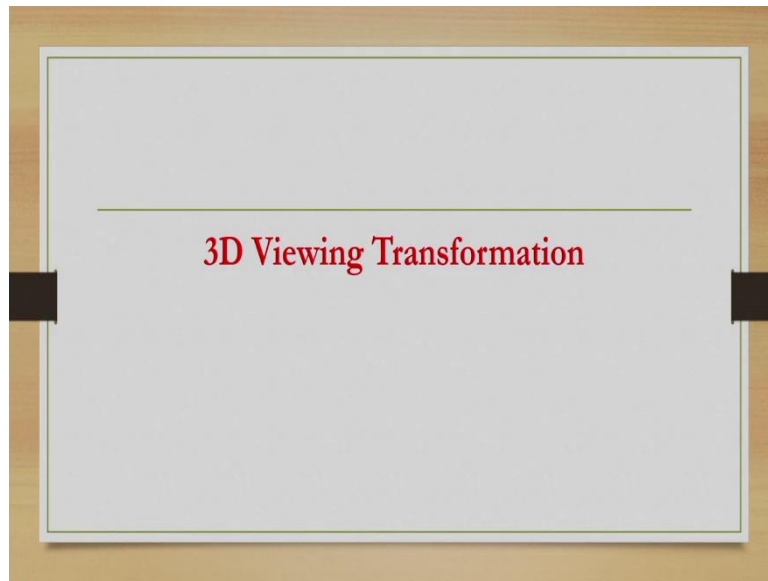


Now, from the view plane after projection, we perform another transformation. The projection is done on a viewport on the view plane. Now, from there, we transform the objects into the description on a device coordinate system. So, when we perform projection transformation, we essentially transform on the view plane. Now, that area where the image is projected is called the window.

Now, from this window, we make a further transformation. From the window, the object descriptions are projected onto a viewport which is on the device coordinate system. So, we perform window to viewport mapping. So, this is the third stage of the viewing pipeline that we are constructing in the fourth stage of the graphics pipeline.

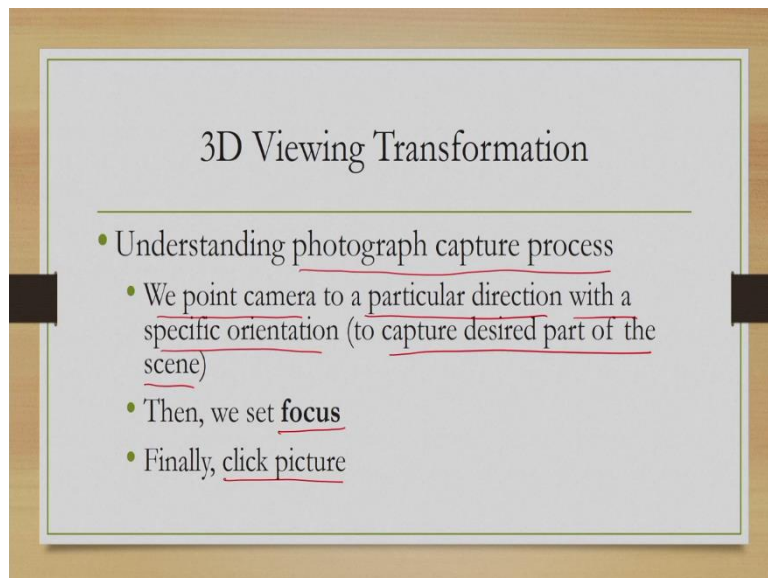
Now, these three are the basic stages in the 3D viewing pipeline. Along with that there are a couple of operations, namely clipping, and hidden surface removal, which together constitute the entire fourth stage of the 3D graphics pipeline, namely the viewing pipeline. And we will discuss each of these sub-stages one by one.

(Refer Slide Time: 09:22)



So, let us start with the first substage that is 3D viewing transformation.

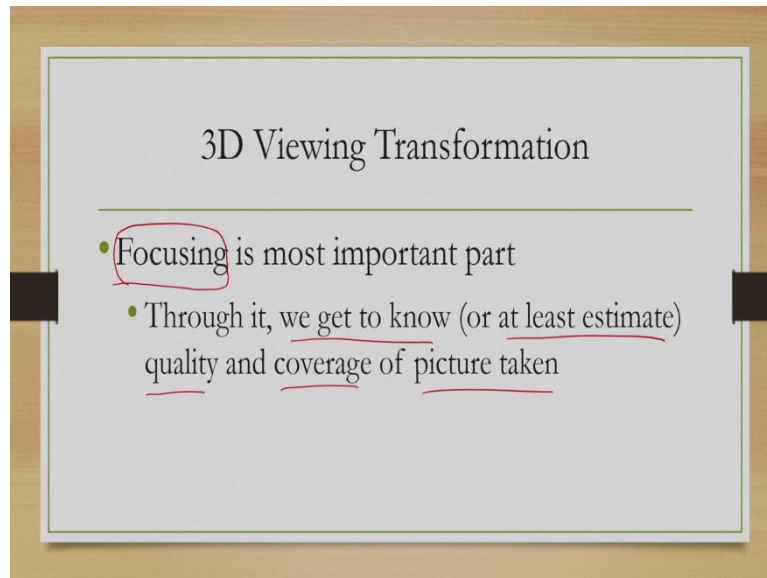
(Refer Slide Time: 09:27)



Now, in order to understand this transformation, we need to understand. How a photograph is taken? So, there are broadly three stages through which we capture a photo. First, we point the camera in a particular direction with a specific orientation so that we can capture the desired part of the scene. Then we set focus, and finally, we click the picture. These three are broadly the stages that we follow when we capture a photo.

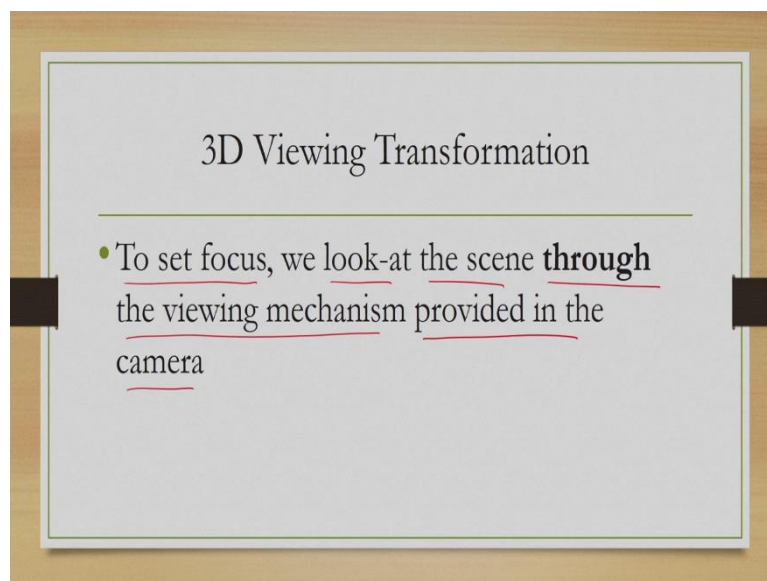


(Refer Slide Time: 10:21)



Now, among the most important thing is focusing. With focusing, we get to know, or at least we can estimate the quality and the coverage of the picture that we are taking. So, focusing constitutes the most important component of the overall process of capturing a photo.

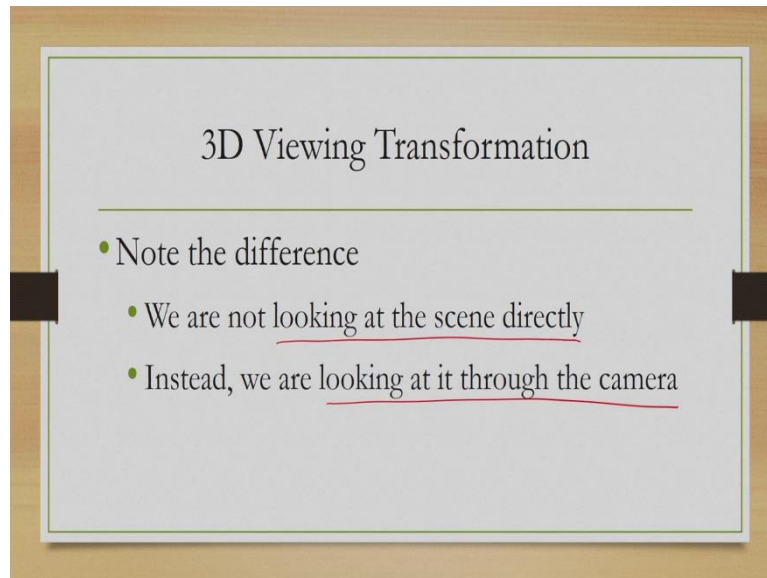
(Refer Slide Time: 10:49)



Now, in order to set the focus, what we do? We essentially look at the scene through a viewing mechanism that is provided in the camera. Typically while we try to set focus, we do not look at the scene with our bare eyes. We look at the scene through the viewing mechanism provided in the camera itself. And accordingly, we set our focus.

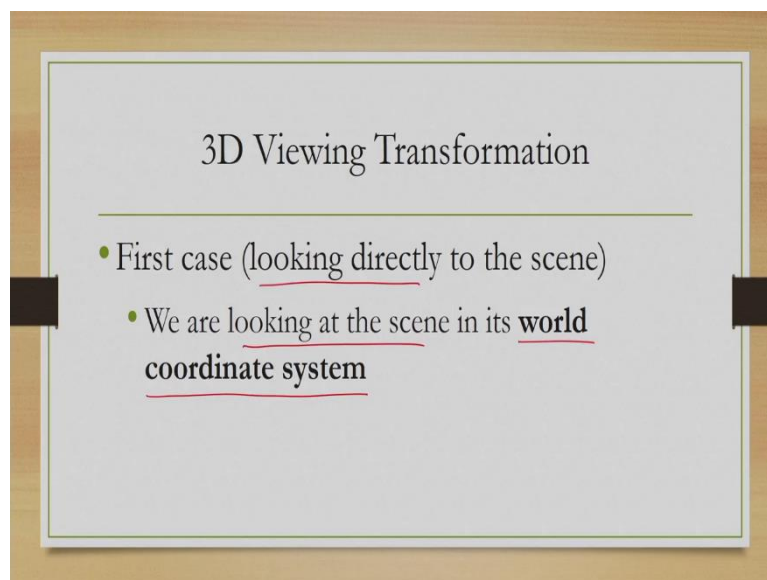


(Refer Slide Time: 11:19)



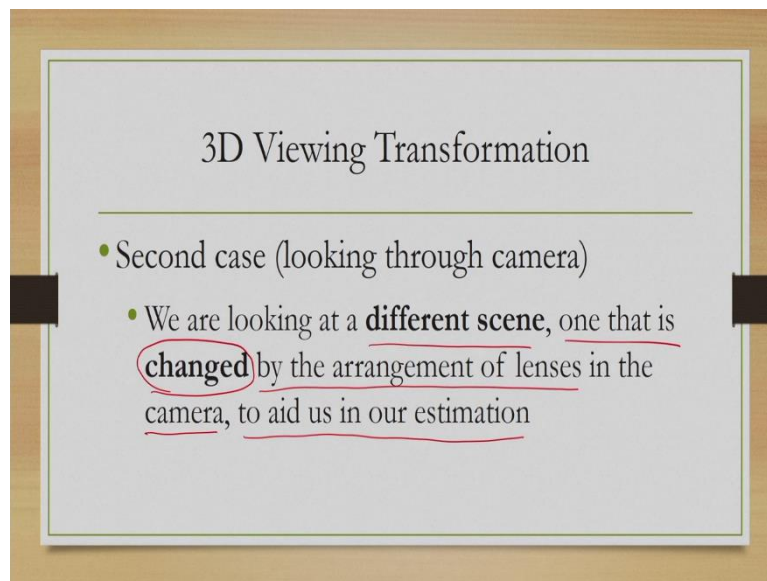
Now, this is important. So, we are not looking at the scene with our bare eyes to set focus. Instead, we are setting focus based on our perception of the scene obtained by looking through a viewing mechanism provided in the camera. So, we are looking at the scene through the camera instead of looking at the scene directly. So, this is very important consideration.

(Refer Slide Time: 11:51)



If we are looking at the scene directly, that means we are looking at the scene in its original coordinate system. That is what we are calling the world coordinate system. So, when we are looking at the scene directly, we are looking at it in its world coordinate reference frame, world coordinate system.

(Refer Slide Time: 12:13)



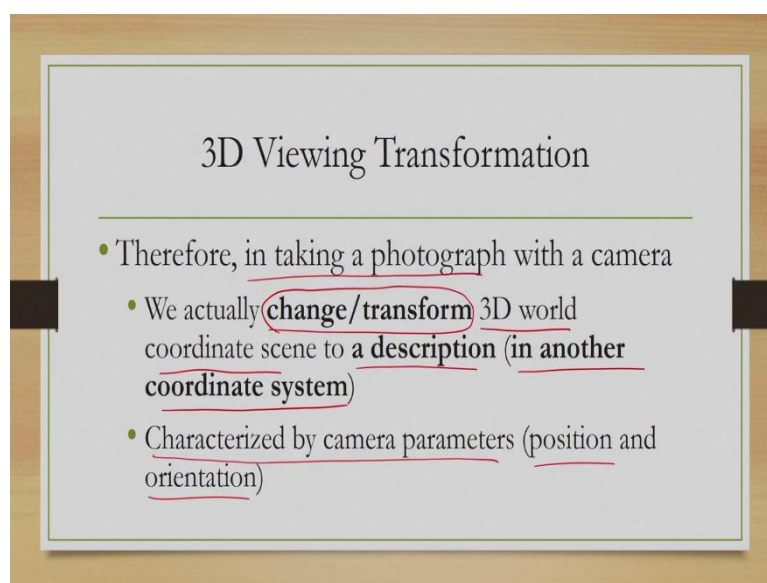
3D Viewing Transformation

- Second case (looking through camera)
  - We are looking at a **different scene**, one that is **changed** by the arrangement of lenses in the camera, to aid us in our estimation

However, if we are looking at the scene through a viewing mechanism of the camera, then we are not looking at the scene in its world coordinate system. Instead, we are looking at a different scene, one that is changed; it is important to note that a change scene and the change took place due to the arrangement of lenses in the camera. So, that we can estimate the quality and coverage of the photo to be taken

So, here we are not looking at a world coordinate scene; we are looking at a scene that is changed from its world coordinate description due to the arrangement provided in the camera viewing mechanism.

(Refer Slide Time: 13:14)



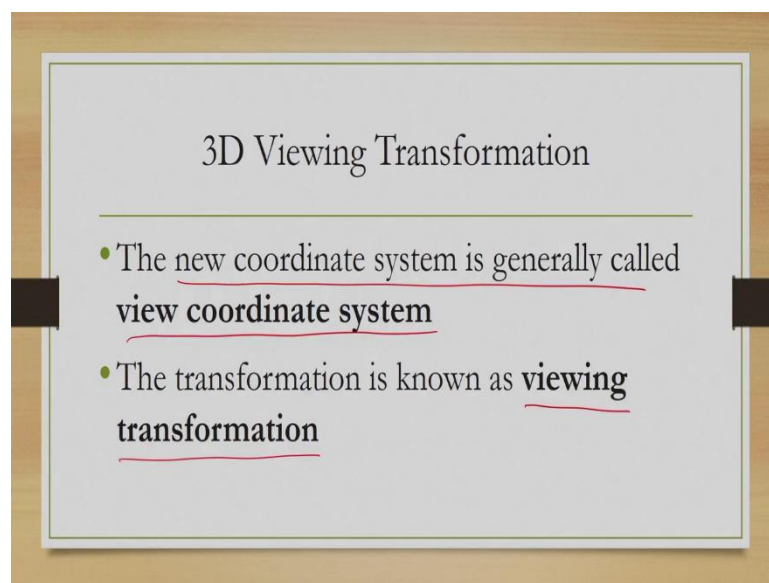
3D Viewing Transformation

- Therefore, in taking a photograph with a camera
  - We actually **change/transform** 3D world coordinate scene to a **description (in another coordinate system)**
  - Characterized by camera parameters (position and orientation)

So, then what happens? When we are taking a photograph with a camera, we are actually changing or transforming the 3D world coordinate scene to a description in another coordinate system. This is the most fundamental concept to be noted to understand how computer graphics simulates the process of taking a photograph.

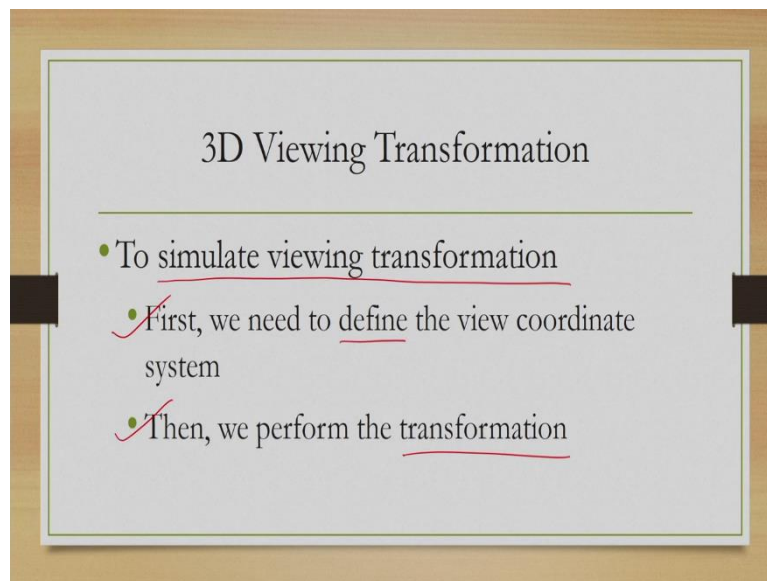
So, when we are looking at a scene to set focus through the viewing mechanism provided in a camera, we are actually transforming the world coordinate scene to a different coordinate system. And this coordinate system is characterized by the camera parameters, namely, the position and orientation of the camera; this needs to be carefully noted.

(Refer Slide Time: 14:18)



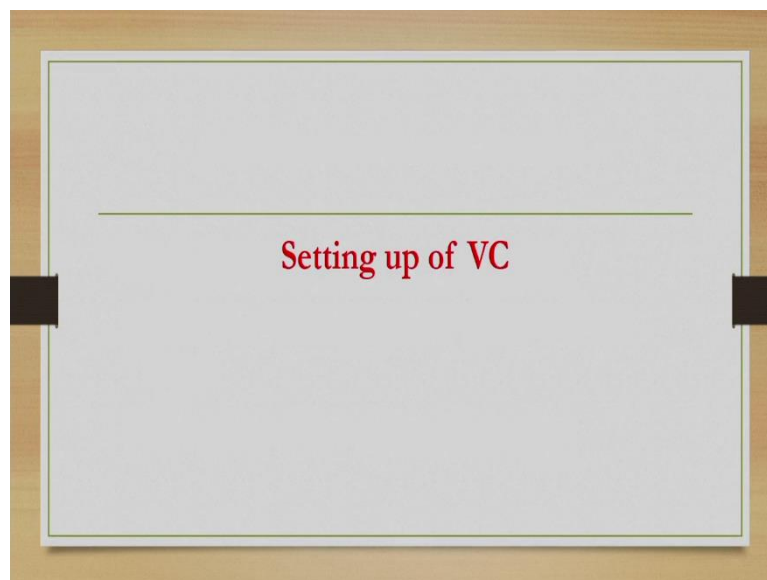
So, this new coordinate system, we generally call it view coordinate system, and the transformation between world coordinate system and view coordinate system is the viewing transformation, which is the first sub-stage of the viewing pipeline. So, essentially we are trying to simulate the photo-taking process, and the first stage in it is to transform the world coordinate description of a scene to the view coordinate system, which simulates the process of looking at the scene through a viewing mechanism provided in the camera.

(Refer Slide Time: 15:11)



So, to simulate this viewing transformation or to implement the viewing transformation we need to do two things. First, we need to define the coordinate system, and second is we perform the transformation. So, first we define the coordinate system, and second, we perform the transformation to simulate this effect of looking at the scene through the camera.

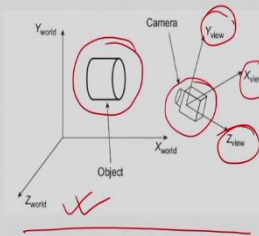
(Refer Slide Time: 15:44)



## View Coordinate Setting Up

- Basic setting shown in figure

- Although we used  $x$ ,  $y$ , and  $z$  in Fig. to denote the three basis vectors of VC, they are usually denoted by  $u$ ,  $v$  and  $n$
- In subsequent discussion, we shall use the latter notation
- How do we determine the VC?



Now, let us go one by one. First, we try to understand how we set up the viewing coordinate system. This figure shows the basic setting that will be considering here. So, on the left side, here is the actual world coordinate scene, so this cylinder is the object in a world coordinate description defined by the three principle axes X, Y and Z.

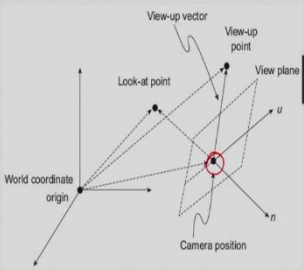
And then this is the camera through which we are looking at the scene, and this view coordinate is characterized by the three principle axes X view, Y view and Z view. Although the more common notation used in graphics is  $u$ ,  $v$  and  $n$  to denote these three principal axes of the view coordinate system rather than  $x$ ,  $y$  and  $z$ .

So, in subsequent discussion, we will refer to this view coordinate system in terms of this latter notation, that is in terms of the principal axes  $u$ ,  $v$  and  $n$ . So, the question is, how do we determine this principal axes, which defines the viewing coordinate system. You may note here that  $n$  corresponds to  $z$ ,  $v$  corresponds to  $y$ , and  $u$  corresponds to  $x$ .

(Refer Slide Time: 17:31)

### View Coordinate Setting Up

- First thing is to determine the origin, where three axes meet
  - We assume camera is represented as a point in the 3D world coordinate system
  - We simply choose this point as our origin (denoted by  $o$ )



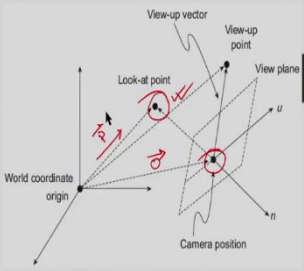
The diagram shows a 3D world coordinate system with axes  $u$ ,  $v$ , and  $n$ . The origin is labeled 'World coordinate origin'. A 'Camera position' is marked with a red circle. A 'View-up vector' is shown as a dashed line from the camera position. A 'View-up point' is marked with a red circle. A 'View plane' is shown as a dashed line. A 'Look-at point' is marked with a red circle. The camera position is also labeled 'Camera position'.

So, let us try to understand how we can determine the three principal axes to define the view coordinate system. So, the first thing is to determine the origin, origin of the view coordinate system where the three axes meet. Now, this is simple; we assume that the camera is represented as a point and the camera position is the origin denoted by  $o$ . So, the position of the camera is the origin where we are assuming that the camera actually is defined as a point, a dimensionless entity.

(Refer Slide Time: 18:20)

### View Coordinate Setting Up

- When we try to focus with our camera, we choose a point (in the world coordinate)
  - This is center of interest or look-at point ( $p$ )



The diagram is similar to the previous one, but with a 'Look-at point' marked with a red circle and labeled 'Look-at point'. A red arrow points from the camera position to the look-at point. The look-at point is also labeled 'Look-at point'.

Now, when we try to focus our camera, we choose a point as we have already mentioned before in the world coordinate system, and we call this the center of interest or look at point, as shown here. So, this is our camera position, and this one is the look at Point. As of now,



you may note that both are defined in the world coordinates scene. So, with these points, we can define vectors. So, this will be the origin vector, and this will be the look at point vector  $p$ .

(Refer Slide Time: 19:10)

### View Coordinate Setting Up

- Using simple vector algebra, we can see  $\hat{n} = \frac{\vec{o} - \vec{p}}{|\vec{o} - \vec{p}|}$
- We normalize  $\vec{n}$  to get the unit basis vector  $\hat{n} = \frac{\vec{n}}{|\vec{n}|}$

Then using vector algebra, what we can do is we can define  $n$  as you can see,  $n$  is the normal of the plane. So, we can define  $n$  to be  $\vec{o} - \vec{p}$  where each of these are vectors? That is the simple vector algebra we can use, and then we normalize  $n$  to get the unit basis vector  $\hat{n}$ , which can be defined simply as  $\frac{n}{|n|}$ . So, then we got one basis vector  $\vec{n}$ .

(Refer Slide Time: 20:00)

### View Coordinate Setting Up

- Next, we specify an arbitrary point  $p_{up}$  along the direction of our head while looking through the camera
- This is the view-up direction

Next, we specify an arbitrary point. Let us denote it by  $p_{up}$  along the direction of our head while looking through the camera. This we call the view of point along the view-up

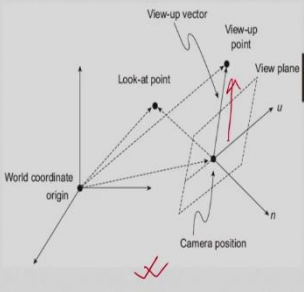


Direction. So, the direction along which our head is oriented while we are looking at the scene through the camera, essentially this is the head up direction.

(Refer Slide Time: 20:37)

### View Coordinate Setting Up

- With this point, we determine view-up vector  $\vec{V}_{up} = \vec{p}_{up} - \vec{o}$
- Then, we get unit basis vector

$$\hat{v} = \frac{\vec{V}_{up}}{|\vec{V}_{up}|}$$


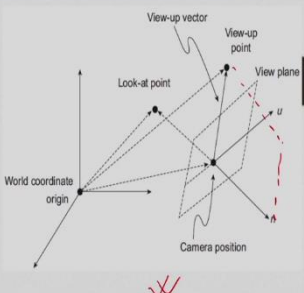
The diagram illustrates the setup of a view coordinate system. It shows a 3D coordinate system with axes. A 'World coordinate origin' is marked. A 'Camera position' is also marked. A 'Look-at point' is shown. A 'View-up point' is shown. A 'View plane' is shown. A 'View-up vector' is shown. A 'View plane' is shown. Unit basis vectors  $u$ ,  $n$ , and  $v$  are shown. A red checkmark is at the bottom.

Now, with this point we determine the view-up vector. This vector  $\vec{V}_{up}$  as a difference of these two vectors as you can see from the figure here. This is again simple vector algebra. And once we get this  $\vec{V}_{up}$  vector, then we can get the unit basis vector  $\hat{v}$  in the same way by dividing the vector with its length, which is a scalar quantity.

(Refer Slide Time: 21:17)

### View Coordinate Setting Up

- We know unit basis vector  $\hat{u}$  is perpendicular to the plane spanned by  $\hat{n}$  and  $\hat{v}$
- Hence,  $\hat{u} = \hat{v} \times \hat{n}$
- Since both  $\hat{n}$  and  $\hat{v}$  are unit vectors, further normalization not required



The diagram illustrates the setup of a view coordinate system. It shows a 3D coordinate system with axes. A 'World coordinate origin' is marked. A 'Camera position' is also marked. A 'Look-at point' is shown. A 'View-up point' is shown. A 'View plane' is shown. A 'View-up vector' is shown. Unit basis vectors  $u$ ,  $n$ , and  $v$  are shown. A red checkmark is at the bottom.

Now, we got two vectors then, two basis vectors  $n$  and  $v$ . Now, if we look at the figure, we can see that the remaining vector  $u$  is perpendicular to the plane that is spanned by  $n$  and  $v$ . So, if this is the plane then  $u$  will be perpendicular to this plane. Then we can simply use the

vector algebra again to define  $u$  to be a cross product of  $v$  and  $n$ ,  $v \times n$ . Now, since both  $n$  and  $v$  are unit vectors? So, further normalization is not required, and we get the unit basis vector  $u$  by this cross product.

(Refer Slide Time: 22:24)

**Summary**

- We are given camera parameters {
  1. Camera position (origin of the coordinate)  $\vec{o}$
  2. View-up point  $\vec{p}_{up}$
  3. Center of interest or the look-at point  $\vec{p}$
- From these, we set up VC
  1. Determine  $\hat{n}$  (unit basis vector opposite to the looking direction):  $\vec{n} = \vec{o} - \vec{p}$ ;  $\hat{n} = \frac{\vec{n}}{|\vec{n}|}$
  2. Determine  $\hat{v}$  (unit basis vector along the view-up direction):  $\vec{V}_{up} = \vec{p}_{up} - \vec{o}$ ;  $\hat{v} = \frac{\vec{V}_{up}}{|\vec{V}_{up}|}$
  3. Determine  $\hat{u}$  (the third unit basis vector):  $\hat{u} = \hat{v} \times \hat{n}$

So, then in summary, what we have done? We assume that few things are given three things; one is the camera position or the coordinate from where we can define the origin vector  $o$ . Then view-up point and corresponding view-up vector we can define and finally the look at point  $p$  and the corresponding vector.

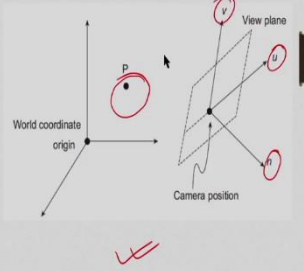
Then based on this information, we perform a three-step process. First, we determine the unit basis vector in  $\hat{n}$  using simple vector algebra, then we determine  $\hat{v}$  again using simple vector algebra. And finally, we determine  $u$  as a cross product of the earlier two vectors that we have defined in the first two steps. And following these stages, we get the three basis vectors that define our viewing coordinate system.

Now, once the coordinate system is defined, our next task that is the second part of the process is to transform the object definition from the world coordinate system to the view coordinate system. Let us see how we can do that. So, in order to transform, we need to perform some operations.

(Refer Slide Time: 24:07)

### Transformation

- Look at the figure illustrating the idea
- **P** is an arbitrary point in WC - we transform to VC



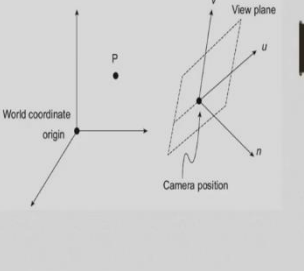
✓✓

To get an idea, let us have a look at the figure here. So, suppose this is an arbitrary point in the world coordinates scene, and we want to transform it to the view coordinate system defined by three vectors  $n$ ,  $u$ , and  $v$ .

(Refer Slide Time: 24:33)

### Transformation

- Assume VC origin (the camera position) at  $(O_{vx}, O_{vy}, O_{vz})$
- 3 basis vectors represented as
  - $\mathbf{u}(u_x, u_y, u_z)$
  - $\mathbf{v}(v_x, v_y, v_z)$
  - $\mathbf{n}(n_x, n_y, n_z)$

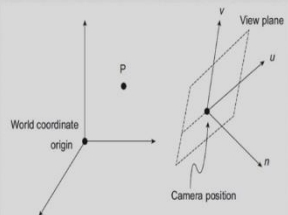


Now, let us assume that the origin is defined with this point having the coordinates then sent here, and the three basis vectors are represented as shown here. These are the X, Y, and Z components of the basis vectors. And this representation will follow to formulate our mechanism to transform any arbitrary point in the world coordinate scene to a point in the view coordinate system.

(Refer Slide Time: 25:12)

### Transformation

- We need transformation matrix  $M_{w2v}$
- When we multiply P with M, we get transformed point (in VC)  
$$P' = M_{w2v} \cdot P$$



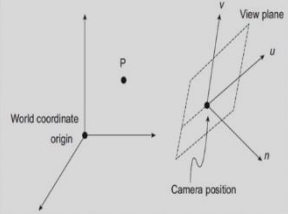
So, what do we need? We need transformation matrix M; if you recollect in the modeling transformation stage, we said that any transformation could be represented in the form of a matrix. So, we need to find out that matrix, which will transform a given point to a point in the view coordinate system.

And how we do that? Again if you recollect our discussion from the lectures on modelling transformation, what we did, we multiply the point with the transformation matrix to get the transformed point. So, this is the transformed point which we will get by multiplying the original point with the transformation matrix.

(Refer Slide Time: 26:03)

### Transformation

- For that, we need to find the **sequence of transformations** required to **align VC with WC**
- We can achieve this by two operations: translation and rotation.



✓✓

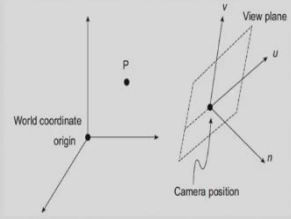
And this transformation is actually a sequence of transformations that are required to align the view coordinate with the world coordinate. In a most general setting they are not aligned like in the figure shown here; there is a slight difference in orientation between the two-coordinate system. So, we align them and then perform the transformation.

Now, in order to do that, we require two basic transformation operations. Translation and rotation, the idea is simple. So, we translate the origin to the world coordinate origin and then rotate the system to align with the world coordinate system.

(Refer Slide Time: 27:04)

### Transformation

- We can do so with TWO operations (transformations)
  - Translation
  - Rotation

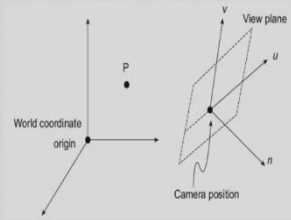


So, this translation and rotation will constitute the sequence of operations we need to transform between the two coordinate systems.

(Refer Slide Time: 27:19)

### Transformation

- First - translate VC origin to WC origin

$$T = \begin{bmatrix} 1 & 0 & 0 & -o_{vx} \\ 1 & 0 & 0 & -o_{vy} \\ 1 & 0 & 0 & -o_{vz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$


Now, first thing is we translate VC origin to world coordinate origin. And this is the transformation matrix, which is the same as we discussed earlier with the corresponding X, Y, Z values replaced here.

(Refer Slide Time: 27:38)

### Transformation

---

- Next - rotate VC to align with WC

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next is the rotation. Now, the rotation matrix is shown here; we will skip the derivation, can be derived. But for the time being, let us just note the rotation matrix. So, this matrix will align if applied this matrix rotates the viewing coordinate system to align it with the world coordinate system.

(Refer Slide Time: 28:09)

### Transformation

---

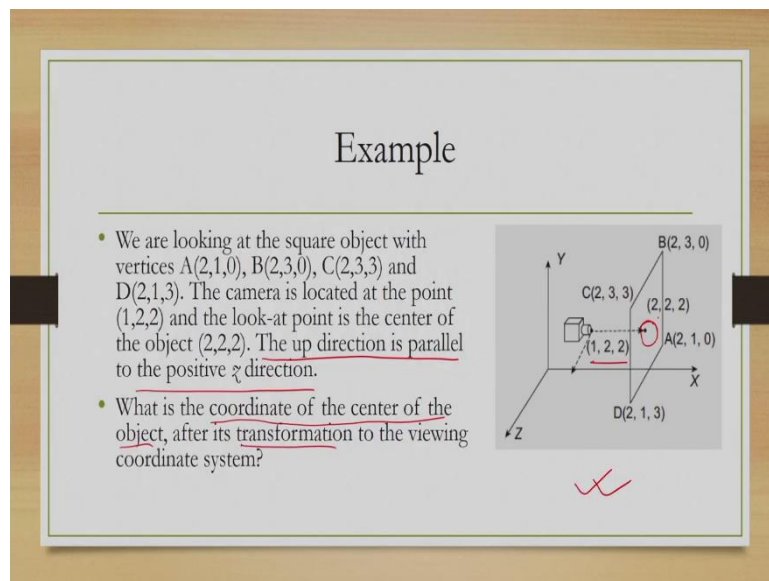

$$M_{w2v} = \underline{R.T}$$

$$P' = \underline{(R.T)P}$$

And since we performed first translation and then the rotation, so we will follow the right to left rule to combine the two transformations to come up with a composite transformation

matrix. Thus we will have to write them in this sequence T first, and then on the left side is R, and we take the product of these two matrices to get the composite matrix. And then we multiply this matrix to the point to get the transformed point coordinates.

(Refer Slide Time: 28:51)



Let us try to understand this process in terms of one example. Consider this setting, here there is a square object defined with its vertices A, B, C, D and then we have a camera located at this point (1, 2, 2) and the look at point is centre of the square object here that is (2, 2, 2). It is also specified that the up direction is parallel to the positive Z direction.

Then given this specification, let us try to calculate the coordinate of the centre of the object after transformation to the view coordinate system. So, originally in the world coordinate



system it is (2, 2, 2). Now, after transformation, what will be the coordinate? Let us try to follow the steps that we have just discussed.

(Refer Slide Time: 30:07)

### Example

- First, we determine 3 unit basis vectors for the viewing coordinate system

The first thing is we determine the 3 unit basis vectors for the viewing coordinate system.

(Refer Slide Time: 30:20)

### Example

- The camera position  $\mathbf{o}$  is (1,2,2)
- Look-at point  $\mathbf{p}$  is (2,2,2)
- Therefore  $\mathbf{n} = \mathbf{o} - \mathbf{p}$   

$$= (-1, 0, 0)$$

$$= \hat{\mathbf{n}}$$

Now, the camera position is defined  $\mathbf{o}$  that is (1, 2, 2) as you can see here. Look at point  $\mathbf{p}$  is defined at the centre of the object that is (2, 2, 2). So, then we can calculate the vector  $\mathbf{n}$  as  $\mathbf{o} - \mathbf{p}$  which is (-1, 0, 0). Now, it is already a unit vector, so no need to do any further operations. So, we already got the unit basis vector  $\hat{\mathbf{n}}$ .

(Refer Slide Time: 31:00)

### Example

- It is already mentioned up direction is parallel to the positive  $z$  direction
  - We can directly determine that  $\hat{v} = (0, 0, 1)$  without any further calculations
- Another way of specifying up vector (instead of specifying point in up direction)

Now, it is also mentioned that the up direction is parallel to the positive Z Direction. Therefore, we can directly determine that the unit basis vector along the up direction  $\hat{v}$  is basically the unit basis vector along the z direction only, that is  $(0, 0, 1)$ , we do not need to do any further calculations.

So, this as you can see is another way of specifying the up vector you tell the direction in terms of available basis vectors or in terms of a line rather than specifying a point. So, there are different ways of specifying the up direction. Anyway, so we have already found out two basis vectors  $\mathbf{n}$  and  $\mathbf{v}$ .

(Refer Slide Time: 32:05)

### Example

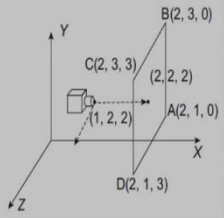
- Finally, cross product of the two vectors (i.e.,  $\hat{v} \times \hat{n}$ ) gives us
  - $\hat{u} = (0, 1, 0)$

Then we take a cross product of these two basis vectors to get the third basis vector  $\hat{u}$ , which is (0, 1, 0).

(Refer Slide Time: 32:16)

### Example

- After 3 basis vectors are determined, we compute transformation matrix  $M_{w2v}$
- Composition of translation and rotation (i.e., R.T)

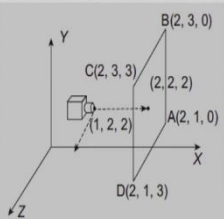


So, then we found the view coordinate system as defined by the three-unit basis vectors  $n$ ,  $u$ , and  $v$ . Next, we compute the transformation matrix  $M$  which is the composition of the translation and rotation matrices.

(Refer Slide Time: 32:45)

### Example

- Camera is at (1,2,2)
- Hence,  $T = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



Now, we have already noted earlier that the translation matrix is represented in this way where we use the coordinate position of the origin since it is already given to be (1, 2, 2) so we replace it here to get the translation matrix in this form. Again, we already know the rotation matrix, which is in terms of the vectors which define the coordinate system, and we

already have determined this vector. So, we replace those values here. So,  $\hat{n}$  is  $(-1, 0, 0)$ ;  $\hat{u}$  is  $(0, 1, 0)$  and  $\hat{v}$  is  $(0, 0, 1)$  we already have determined this. Now, we replace these values. This is for  $u$ , this is for  $v$ , and this is for  $n$  to get this rotation matrix  $R$ .

(Refer Slide Time: 34:14)

Example

$$M_{w2v} = R.T$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & -2 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, then we multiply these to  $R$  dot  $T$  to get the transformation matrix  $m$  shown here.

(Refer Slide Time: 34:29)

Example

- New coordinates of object center  $(2,2,2)$  is,

$$\hat{C} = M_{w2v} \hat{C} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 1 & 0 & 0 & -2 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

- i.e., center gets transformed to  $(0,0,-1)$  in VC

Now, we have determined  $M$ , then we multiply  $M$  with the original coordinate to get the transformed coordinate and note here that it will be in the homogeneous coordinate system. But with a homogeneous factor 1, so we do not need to make any change. So, after multiplication, what we get? We get that the coordinate of the transformed point is  $(0, 0, -1)$

in the view coordinate system. So, that is our transformed point in the view coordinate system.

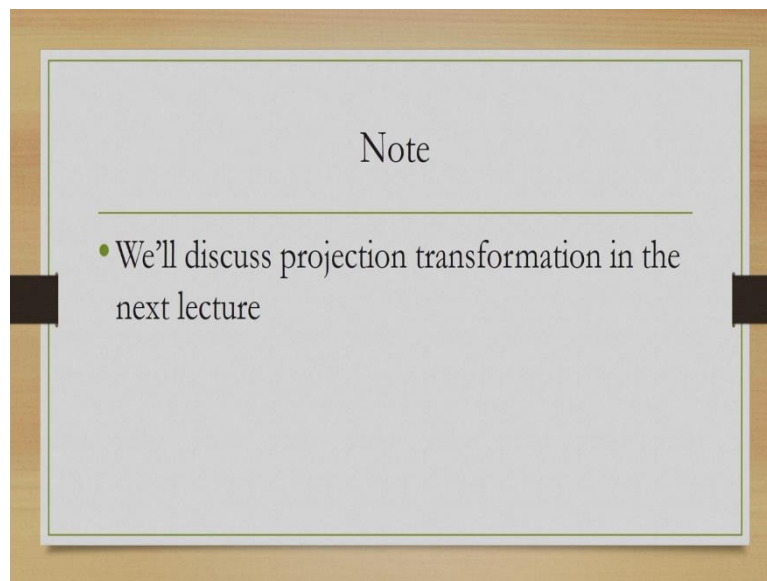
So, in summary, today what we discussed? So, we are discussing the fourth stage that is viewing pipeline which is essentially simulating the process of capturing a photo. Now, this process consists of multiple stages, broadly, there are three stages. First is a transformation from the world coordinate description of an object to a view coordinate system. The second is from view coordinate system to projection on a view plane, and the third is from the view plane a transformation to the device coordinate system.

Among them we discussed the first major stage that is the transformation from world coordinate description to a view coordinate description. There we have seen how we can define a view coordinate system in terms of its three-principle axis  $u$ ,  $v$ , and  $n$  and how to determine these three principal axes given the information of the camera position, the view of vector, and the look at point.

Once these three are given we can define the three principal axes or the view coordinate system, which in turn gives us the system itself. Then once the system is defined, we determine a transformation matrix, which is a composition of translation and rotation to transform a point from the world coordinate system to the view coordinate system.

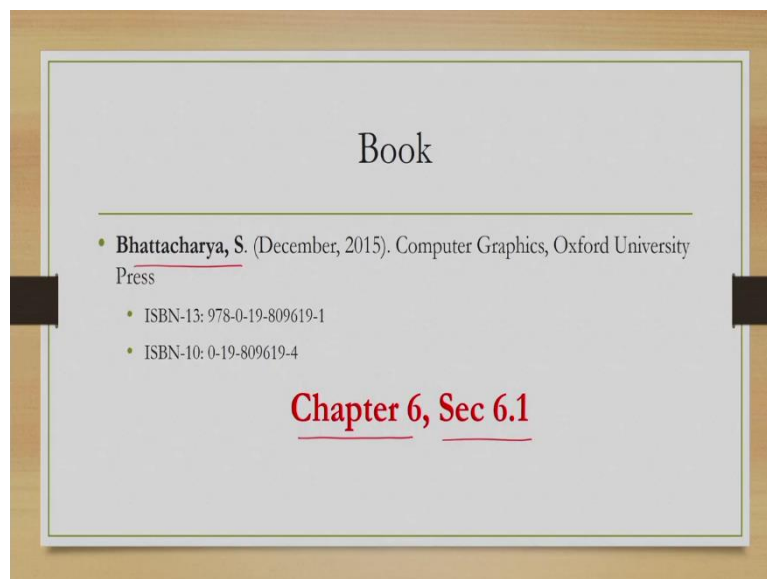
We achieve this by multiplying the world coordinate point with the transformation matrix to get the transformed point in the view coordinate system. We may note here that here also we are assuming a homogeneous coordinate system. However, the homogeneous factor is still 1, so we do not need to do any further change in the computed coordinate value.

(Refer Slide Time: 37:35)



In the next lecture, we will talk about the second major stage in this viewing pipeline, namely, the projection transformation.

(Refer Slide Time: 37:47)



Whatever we have discussed today can be found in this book. And you are advised to refer to chapter 6, section 6.1, to learn in more detail the topics that we have covered today. Thank you, and goodbye.