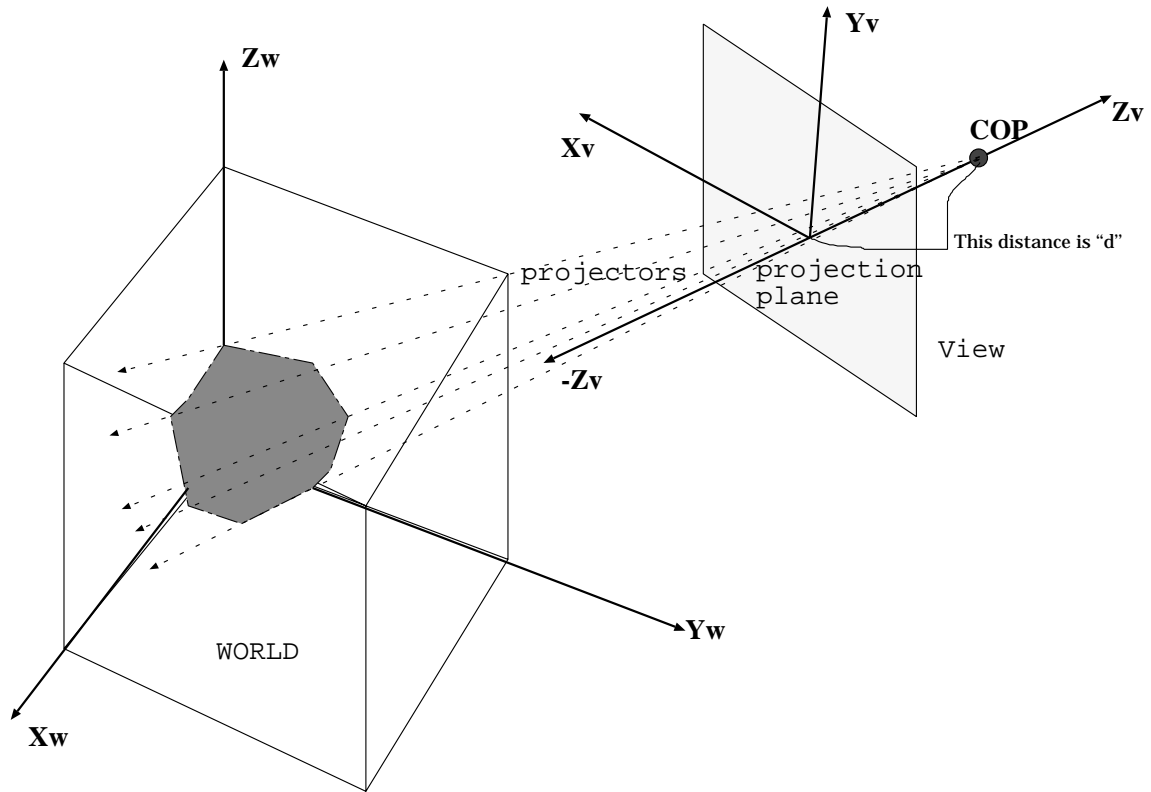


- Steps for 3D Viewing are similar to the process involved in taking a snapshot.
- It involves: i) positioning the camera, ii) camera orientation, i.e, which way do we point the camera and how should we rotate it around the line of sight to set up the up direction of the picture? and finally, iii) the scene is cropped to the size of the “window” (aperture of the camera) and light from the visible surfaces is projected onto the camera film.
- Modeling coordinates — World coordinates — Viewing coordinates — Projection coordinates — Device coordinates
- Viewing coordinate system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane.

- **Viewing transformations** are different from **Modeling transformations**.
- Modeling transformations represent the object position in the world, and Viewing transformations are applied to map the object's position from world to the screen.
- Viewing transformations do not change the (configuration of the) actual world in any fashion.
- To establish the viewing-coordinate reference frame, first pick a world coordinate position called the **view reference point**. This point is the origin of our viewing-coordinate system.
- We then select the positive direction for the viewing  $Z_v$  axis (in OpenGL, default is the -z axis), and the orientation of the view plane, by specifying the **view-plane normal vector,  $\mathbf{N}$** . OpenGL establishes the direction for  $\mathbf{N}$  using the selected coordinate position as a *look-at point* relative to the VRP. Direction of  $\mathbf{N}$  is from the look-at point to the VRP.
- Finally, we choose the up direction for the view by specifying a vector  $\mathbf{V}$ , the **view-up vector**. User chooses any convenient direction (not-parallel to  $\mathbf{N}$ ).



Specify:

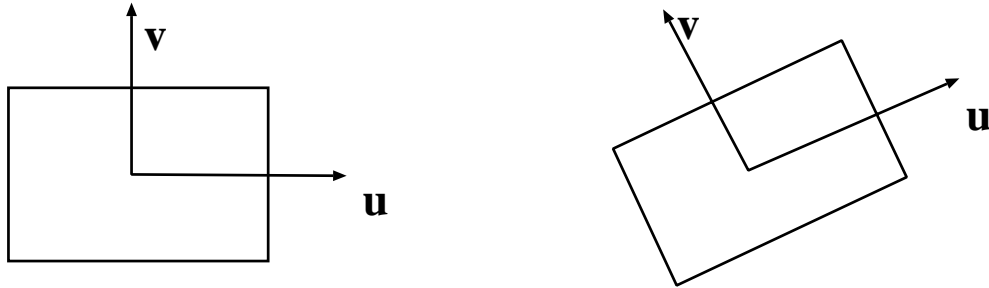
Position of Viewing (VRP)

Direction of Viewing (View Direction)

Picture's Up-Direction (upVector)

All the specifications are in World Coordinates.

Upvector denotes the orientation of the View window on the View reference plane.



Once the VRC is defined, the 3D objects are clipped and projected on to the VRP.

One simple situation is when VRP coincides with the world origin and  $u, v, n$  is nothing but  $x, y, z$ .

Simple projection could be used (such as ignoring  $z$ ).

- World should be transformed so that the **view reference coordinate system** coincides with the **world coordinate system**.
- This transformation is called **world to viewing coordinate transformation**.
- the transformation matrix is called **view orientation matrix**.

## Steps (without using OpenGL)

1. Translate the view reference point to the origin of the world-coordinate system.
2. Apply rotations to align the  $x_v$ ,  $y_v$ , and  $z_v$  axes with the world  $x_w$ ,  $y_w$ , and  $z_w$  axes, respectively.

If the view reference point is specified at world position  $(x_0, y_0, z_0)$ , this point is translated to the world origin with the matrix transformation

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_1, n_2, n_3)$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_1, u_2, u_3)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_1, v_2, v_3)$$

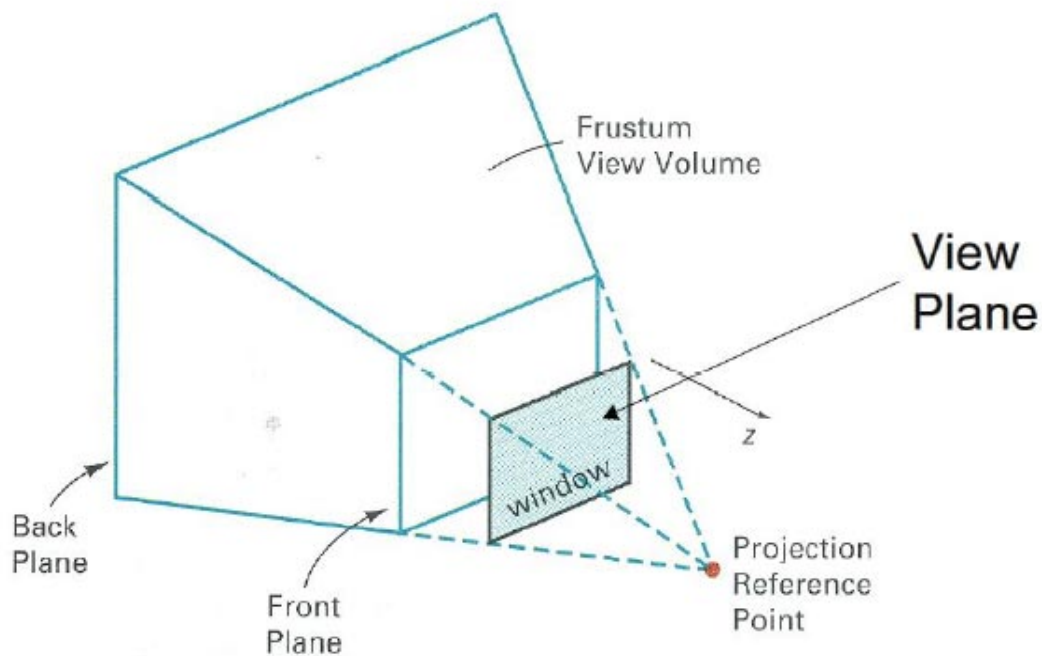
- This method will automatically adjust the direction for  $\mathbf{V}$  so that  $\mathbf{v}$  is perpendicular to  $\mathbf{n}$ .

The composite transformation matrix for the viewing transformation is  $\mathbf{RT}$  where,

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

## View Window

- View Window is a rectangle in the view plane specified in terms of view coordinates
- Specify center  $(cx, cy)$ , width and height
- prp lies on the axis passing through the center of the view window and parallel to the n-axis.
- prp is cop



## Complete View Specification

- Specification in World Coordinates: i) position of viewing (vrp), direction of viewing ( $-\mathbf{n}$ ), up direction for viewing (**upVector**)
- Specification in view coordinates: i) view window - center (cx,cy), width and height.
- prp: distance from the view plane
- front clipping plane: distance from view plane
- back clipping plane: distance from view plane

## Parallel Projection

1. Apply the world to view transformation
2. Apply the parallel projection matrix to project the 3D world onto the view plane

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

3. Apply 2D viewing transformation to map the view window to the screen

## Perspective Projection

1. Apply the world to view transformation
2. Apply translation, such that the center of the view window coincide with the origin
3. Apply the perspective matrix to project the 3D world onto the view plane

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -cx \\ 0 & 1 & 0 & -cy \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

4. Apply 2D viewing transformations to map the view window (centered at the origin) onto the screen.