# Implementing a virtual camera

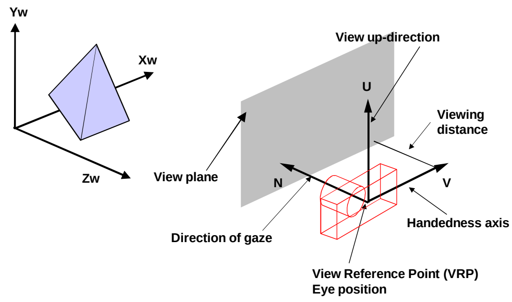
Defining a camera

Implementing a virtual snapshot

- Coordinate system transformations
- Viewing projections

Practical exercise

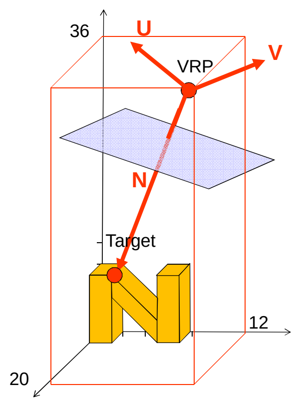
# Virtual camera - definitions




# Creating a view of the scene – an outline

1. Create vertex tables (3D) for an object in the World coordinate system.
2. Define the (3D) Viewing (camera) coordinate system.
3. Change the 3D coordinates of the object from the World system to the Viewing system.
4. Create (2D) perspective projection of the object.
5. Plot the 2D vertices, edges and surfaces.

# 2. Define the camera coordinate system




# 2. The camera coordinate system

- Define N

  N=TP-VRP

- Define a temporary “up-vector”

  $U_0=[0\ 1\ 0]$

- Compute the handedness vector

  $V=U_0 \times N$

- Compute the correct up-vector

  $U=N \times V$

# 3. Coordinate system transformation

3.1.

Define a matrix to translate the centre of the Camera system to the centre of the World system

$$TM = \begin{bmatrix} 1 & 0 & 0 & -x_{VRP} \\ 0 & 1 & 0 & -y_{VRP} \\ 0 & 0 & 1 & -z_{VRP} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 3. Coordinate system transformation

3.2

Define a matrix to align the axes of the Camera system with the axes of the World system

$$R_{xyz} = \begin{bmatrix} \frac{V_x}{|\mathbf{V}|} & \frac{V_y}{|\mathbf{V}|} & \frac{V_z}{|\mathbf{V}|} & 0 \\ \frac{U_x}{|\mathbf{U}|} & \frac{U_y}{|\mathbf{U}|} & \frac{U_z}{|\mathbf{U}|} & 0 \\ \frac{N_x}{|\mathbf{N}|} & \frac{N_y}{|\mathbf{N}|} & \frac{N_z}{|\mathbf{N}|} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

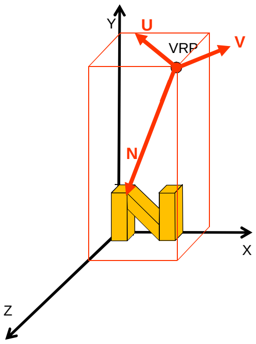
## 3. Coordinate system transformation

3.3.

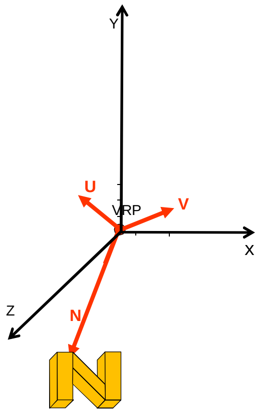
Define a matrix to convert from the right-handed World coordinate system to the left-handed Camera coordinate system (scaling by -1 w.r.t. X)

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

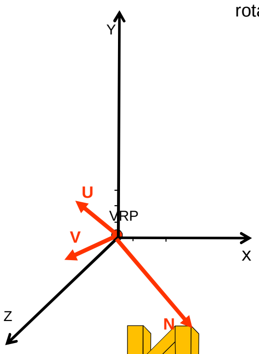
W – world coordinate system
C – camera coordinate system



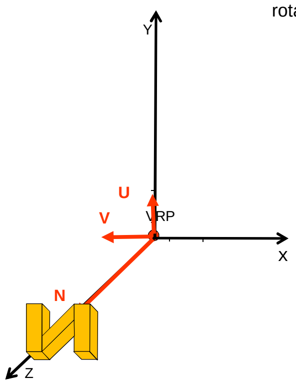

Translate the origin of C to the origin of W



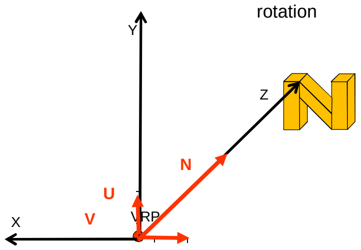

Align the axes U and N of C to the axes Y and Z of W by rotation ...



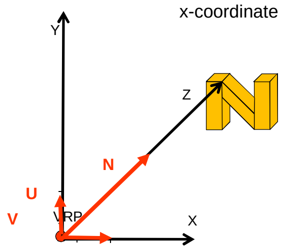

... Align the axes U and N of C to the axes Y and Z of W by rotation ...



... Align the axes U and N of C to the axes Y and Z of W by rotation




Align the axis V of C to the axis X of W by axis “flipping” (scaling x-coordinate by -1)




## 4. Perspective projection

Define a perspective projection matrix with Centre of Projection (COP) at the centre of the coordinate system and the viewing plane at the distance D

$$P_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/D & 0 \end{bmatrix}$$

Perspective projections are covered in unit ‘Viewing transformations’

## 5. Transform the object vertices

- Vertices defined in a N x 4 matrix P
- Compute the combined transformation matrix
  $CM = SM * P_{per} * R_{xyz} * TM$
- Apply the combined transformation matrix to the vertices
  $P' = CM * P$

Composite transformations are covered in unit ‘Composite transformations’

## 6. Homogeneous coordinates

- Perspective projection makes the vertices P’ non-homogeneous.
- Make them homogenous by dividing the coordinate vector of each vertex by its fourth (homogeneous) component

$$P' = \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \qquad P'' = \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} / h = \begin{bmatrix} x/h \\ y/h \\ z/h \\ 1 \end{bmatrix}$$

## 7. Plot the 2D points

- The ‘z’ coordinate for all the homogeneous vertices should be equal D
- Plot the object in 2D, using only (x,y) coordinates of the homogenous vertices, and the object’s surface table