3D Viewing
– Transform the 3D modeling coordinates to 2D Device coordinates

The aspects to be considered:

• View position and orientation
  – we can view the object from arbitrary spatial position.

• Projection
  – The 3D object must be projected onto a view plane

• View volume and Clipping
  – The clipping boundaries encloses a volume of space.
Transformation for 3D view

Figure: 3D Viewing Transformation pipeline
(Xw, Yw, Zw) is the world coordinate system

(Xv, Yv, Zv) is the viewing coordinate system

Right-hand viewing coordinate system

P0 is the view reference point
P0 is also the camera position, origin of the view coordinate

(−n is the viewing direction)
(v is the view-up vector)

u, v, n are the unit vectors on Xv, Yv, and Zv axes

(eye looks at origin O)
Four parameters representation in spherical coordinate (phi, theta, L) and d
phi is polar angle
theta is azimuthal angle
L is the distance from O to Eye
d is the distance from eye to the projection plane
Viewing coordinates formation

- View coordinate system (VC: $x_v, y_v, z_v$)
  → Its origin is called view reference point (VRP)

- View plane
  → projection plane which is perpendicular to the viewing $z_v$ axis
  → view plane normal vector (e.g., $N$ vector from look-at point to the view reference point)

- View-up vector (i.e., $V$ vector)
  → Up direction of the view

- $U$ vector
  → Direction for the $x_v$ axis

- $UVN$ system == Viewing system
  → $uv$ plane == $xVy_v$ plane
Viewing coordinates generation

- Translation: VRP \( \Rightarrow \) WC origin

- Rotation: alignment from VC to WC:
  \((x_v, y_v, z_v) \rightarrow (x_w, y_w, z_w)\)

\[
M_{w_c-v_c} = R \cdot T
\]

\[
= \begin{bmatrix}
  u_1 & u_2 & u_3 & 0 \\
  v_1 & v_2 & v_3 & 0 \\
  n_1 & n_2 & n_3 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  1 & 0 & 0 & -x_0 \\
  0 & 1 & 0 & -y_0 \\
  0 & 0 & 1 & -z_0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

where the unit vectors are:

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

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

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

OpenGL command:

\texttt{gluLookAt(eyex, eyey, eyez, lookatX, lookatY, lookatZ, upx, upy, upz)\)

5
Projection under viewing coordinates

Two types of basic projections:

- Parallel projection
  → project the object onto the view plane along the parallel lines

- Perspective projection
  → The projection lines converge to a common center point (called projection reference point (PRP), or center of projection (COP)).
pararrel projection

view plane

Project reference point

pararrel projection
Properties of the projection:

- Parallel projection:
  → Relative proportions of objects are preserved
  → Less realistic

- Perspective projection:
  → The projections of objects depend on the distance between the objects and the view plane
  → Realistic
  → Relative proportions of objects are not preserved.
Projection category:

- Parallel projection
  - Orthographic parallel projection → projection lines (or projection vector) are perpendicular to the view plane → Project point \((x, y, z)\) into the projection coordinates in the view plane: \((x_p, y_p) = (x, y)\)
  - Oblique parallel projection → projection lines are not perpendicular to the view plane → Project a point \((x, y, z)\) into the viewing plane by transform matrix:

\[
M_{\text{parallel}} = \begin{bmatrix}
1 & 0 & L_1 \cos(\phi) & 0 \\
0 & 1 & L_1 \sin(\phi) & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
where

\[ L_1 = \frac{1}{\tan(\alpha)} \]

\( \phi \) is the angle between the projected line and the x axis.

\( \alpha \) is the angle between projection vector and the the projected line.

Note: \( \phi = 60^\circ \): realistic (lines perpendicular to the view plane are projected at 1/2 of their length

- Perspective projection based on the number of principal vanishing points
  - 1-point projection
– 2-point projection
– 3-point projection

Note:

**Vanishing point** is the point at which a set of projected parallel lines appears to converge.

**Principal vanishing point** is the converging point for the set of lines that parallel to the principal axis.
Note: The planar geometric projection is known as the projection is onto a plane rather than onto a curved surface.
**Perspective Projection Matrix**

- projection reference point $z_{prp}$ is along the $z_v$ axis

Perspective projection of a point $P = (x, y, z)$ to position $(x_p, y_p, z_{vp})$ on the view plane:

\[ x_p = x \left( \frac{d_p}{z_{prp} - z} \right) \]
\[ y_p = y \left( \frac{d_p}{z_{prp} - z} \right) \]

where $d_p = z_{prp} - z_{vp}$

Special cases to be considered:

(1) PRP is not along the $z_v$ axis
(2) $z_{vp} = 0$: $uv$ plane is taken as the view plane
(3) $z_{prp} = 0$: Take the PRP as the viewing coordinate origin
Perspective-projection transformation matrix:

\[
\begin{bmatrix}
  x_p \\
  y_p \\
  z_p \\
  1
\end{bmatrix} = \begin{bmatrix}
  x_h \\
  y_h \\
  z_h \\
  h
\end{bmatrix} \cdot \frac{1}{h}
\]

\[
\begin{bmatrix}
  x_h \\
  y_h \\
  z_h \\
  h
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & -z_{vp}/d_p & z_{vp}(z_{prp}/d_p) \\
  0 & 0 & -1/d_p & z_{prp}/d_p
\end{bmatrix} \cdot \begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]

where \( h \) is the homogeneous factor:

\[
h = (z_{prp} - z)/d_p
\]
View Volumes

- View window: the 2D display within the view plane

- View volume: Based on the view window boundaries, a volume of space is obtained for the object display, the objects outside the view volume are clipped. The boundary planes in $z_v$ direction are referred to as the front plane (near plane) and back plane (far plane).

- Frustum: It is formed by truncating the pyramidal view volume for the perspective projection by the front and back clipping planes.

Note:
→ view plane cannot contain the projection reference point (PRP)

→ PRP cannot locate in between the front and back planes

→ View volume and perspective effect can be changed by the positions of the PRP and the window

→ Relative positions of the object and the view plane affect the projected size of the object.
parallel projection

parallelepiped view volume

parallel projection

Frustum view volume

projection reference point

perspective projection

back plane

(far)

front plane

(near)

Zv

near

window

window

far
General Perspective projection transformation

Matrix: \( M_{\text{perspective}} = M_{\text{scale}} \cdot M_{\text{shear}} \)

\( M_{\text{shear}} \): Shear the view volume \( \rightarrow \) align the centerline of the frustum to be perpendicular to the view plane

\( M_{\text{scale}} \): Scale the view volume to a parallelepiped window volume

\[
M_{\text{shear}} = \begin{bmatrix}
1 & 0 & a & -az_{prp} \\
0 & 1 & b & -bz_{prp} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\( a, b \) are the shear parameters:

\[
a = -\frac{x_{prp} - (x_{w\text{min}} + x_{w\text{max}})/2}{z_{prp}}
\]

\[
b = -\frac{y_{prp} - (y_{w\text{min}} + y_{w\text{max}})/2}{z_{prp}}
\]
\[
M_{scale} = \begin{bmatrix}
1 & 0 & \frac{-x_{prp}}{z_{prp} - z_{vp}} & \frac{x_{prp}z_{vp}}{z_{prp} - z_{vp}} \\
0 & 1 & \frac{-y_{prp}}{z_{prp} - z_{vp}} & \frac{y_{prp}z_{vp}}{z_{prp} - z_{vp}} \\
0 & 0 & 1 & 0 \\
0 & 0 & \frac{-1}{z_{prp} - z_{vp}} & \frac{x_{prp}z_{vp}}{z_{prp} - z_{vp}} \\
\end{bmatrix}
\]
3D Clipping

- Clip the object against the boundary planes of the view volume

- clip line segments
  Both end points of the line segment are outside a boundary plane $\Rightarrow$ trivial reject
  Both end points of the line segment are inside ALL boundary planes $\Rightarrow$ trivial accept
  Otherwise, calculate the intersection point between the line and the boundary plane

Definition of a point $(x, y, z)$:

Outside the boundary plane: $Ax + By + Cz + D > 0$
Inside: $Ax + By + Cz + D < 0$
Intersection: $Ax + By + Cz + D = 0$
• clip polygon surface
  Clip all the polygon edges against the boundary plane

In order to transform the 3D viewing coordinates to 2D device coordinates and easily clip in 3D space, we must transform the view volume to the normalized rectangular parallelepiped (normalized view volume), this process is called projection transformation.

The normalized view volume is in the range:

\[ x \in [0, 1] \]
\[ y \in [0, 1] \]
\[ z \in [0, 1] \]

• 3D view volume to 3D viewport transformation
  (3D window \( \Rightarrow \) 3D viewport)
\[ M = \begin{bmatrix}
D_x & 0 & 0 & K_x \\
0 & D_y & 0 & K_y \\
0 & 0 & D_z & K_z \\
0 & 0 & 0 & 1
\end{bmatrix} \]

\[
\begin{bmatrix}
x_v \\
y_v \\
z_v \\
1
\end{bmatrix} = M \cdot \begin{bmatrix}
x_w \\
y_w \\
z_w \\
1
\end{bmatrix}
\]

Scale factor: \( D_x = \frac{x_{v_{\text{max}}}-x_{v_{\text{min}}}}{x_{w_{\text{max}}}-x_{w_{\text{min}}}} \)

Translation: \( K_x = x_{v_{\text{min}}} - x_{w_{\text{min}}}D_x \)

Similarly, \( D_y, D_z, K_y, K_z \) can be defined.
3D viewport clipping

- 3D region code: $-6$ bits

  From right to left, the bit in the region code is defined: bit $1 = 1$, if $x < x_{vmin}$ (left)
bet $2 = 1$, if $x > x_{vmax}$ (right)
bet $3 = 1$, if $y < y_{vmin}$ (below)
bet $4 = 1$, if $y > y_{vmax}$ (above)
bet $5 = 1$, if $z < z_{vmin}$ (front)
bet $6 = 1$, if $z > z_{vmax}$ (back)

  E.g., "101010" $\Rightarrow$ a point above, behind, and right of the 3D viewport.

- Clipping
Region code (A, B) for the two end points of a segment:

\[ A \& B \neq 0 \Rightarrow \text{trivial reject} \]

\[ A = 0 \text{ and } B = 0 \Rightarrow \text{trivial accept} \]

\[ A \& B = 0 \text{ and } A \neq 0 \text{ or } B \neq 0 \Rightarrow \text{calculate the intersection of the line segment and the viewport boundary plane} \]
\[ \Rightarrow \text{check the results to make sure that the intersection points are within the viewport range, and on the line segment (not on the line extension part).} \]