7. 3D Viewing

7.1 Projections:

• Why is projection necessary?

(a) Perspective projection

(b) Parallel projection
Vanishing point

- Perspective projections of parallel lines not parallel to the projection plane will converge to a vanishing point

Principal vanishing point

- vanishing point of a set of parallel lines that is parallel to one of the three principal axes
e.g., three orthographic projections

Isometric projection of a unit cube along the direction \((1, -1, -1)\).
e.g., oblique projection
Various planar projections:

Planar geometric projections

Parallel

Orthogonal
Top
Front
Axonometric
Side
isometric
Other

oblique
Cabinet
Cavilier
Other

Perspective

one-point
two-point
three-point
7.2 Mathematics of Projections

- Projections can be defined by $4 \times 4$ matrices
- Projection plane is normal to the $z$ axis

**Perspective projection:** (not Affine, irreversible)

$$[x_p, y_p, z_p, 1]^t = \left[ \frac{x}{-z/d}, \frac{y}{-z/d}, -d, 1 \right]^t$$

$$= [x, y, z, -z/d]^t$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/d & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= M_{per}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}$$
Parallel (orthogonal) projection:

\[ [x_p, y_p, z_p, 1]^t = [x, y, 0, 1]^t \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix} = M_{par}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
Perspective projection: (COP is not at the origin)

\[ x_p = \frac{x}{1 - z/d} \]
\[ y_p = \frac{y}{1 - z/d} \]
\[ z_p = 0 \]

\[
[x_p, y_p, z_p, 1]^t = \left[ \frac{x}{1 - z/d}, \frac{y}{1 - z/d}, 0, 1 \right]^t
\]

\[ = [0, 0, d, 1]^t + \left[ \frac{x}{1 - z/d}, \frac{y}{1 - z/d}, -d, 1 \right]^t \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & d & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1/d & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z - d \\
1
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -d \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]

\[
= M_t \ M_{per} \ M_t
\]

Oblique projection:

\[
x_p = x - z \cdot \cot \theta
\]
\[
y_p = y - z \cdot \cot \phi
\]
\[ [x_p, y_p, z_p, 1]^t = [x - z \cdot \cot \theta, y - z \cdot \cot \phi, 0, 1]^t \]

\[
\begin{bmatrix}
1 & 0 & -\cot \theta & 0 \\
0 & 1 & -\cot \phi & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & -\cot \theta & 0 \\
0 & 1 & -\cot \phi & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]

\[= M_{par} \cdot M_{shearing} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \]
Where are the vanishing points?

- Parallel lines after perspective projection are still parallel lines if they are also parallel to the projection plane

Why?

If $L(t) = A + c \cdot t$ is a line

$$A = (A_x, A_y, A_z) ; \quad c = (c_x, c_y, c_z)$$

if the view point (eye) is at the origin and the projection plane is perpendicular to the $z$ axis at $-d$, then the perspective projection of $L(t)$ is:

$$L_p(t) = (d \frac{A_x + c_x t}{-A_z - c_z t}, \quad d \frac{A_y + c_y t}{-A_z - c_z t}) \quad (*)$$

If $L(t)$ is parallel to the projection plane ($c_z = 0$) then

$$L_p(t) = \frac{-d}{A_z} (A_x + c_x t, \quad A_y + c_y t)$$

Slope of $L_p(t)$ is $c_y / c_x$. 


Parallel lines after perspective projection are no longer parallel lines if they are not parallel to the projection plane.

Why?

If $L(t)$ is not parallel to the projection plane ($c_z ≠ 0$) then from (*) we that that

$$L_p(t) \rightarrow -d \left( \frac{c_x}{c_z}, \frac{c_y}{c_z} \right) \text{ when } t \rightarrow \infty$$

Hence, any line with the same direction vector would converge to this (vanishing) point

$$-d \left( \frac{c_x}{c_z}, \frac{c_y}{c_z} \right).$$

Principal vanishing point: vanishing point generated by lines parallel to one of the principal axes (at most three PVPs).

Two-point perspective projection is popular.
How to find vanishing points?

Construct a line parallel to $AB$ that passes thru the view point (eye). The intersection of this line with the projection plane is the vanishing point of $AB$. 

$B' \rightarrow VP \text{  when  } B \rightarrow \infty$
7.3 Camera Model for Perspective View

- How to create a perspective view of a scene in OpenGL?
- How to control the camera’s position and orientation in OpenGL?

Conceptual model of 3D viewing:
Define Viewing Coordinate System:
(specification of a 3D view)
(Positioning and pointing the camera)

```c
glMatrixMode ( GL_MODELVIEW );
glLoadIdentity ( );
gluLookAt ( eye.x, eye.y, eye.z, look.x, look.y, look.z, up.x, up.y, up.z);
```

\[
\mathbf{n} = \mathbf{EYE} - \mathbf{LOOK}
\]
\[
\mathbf{u} = \mathbf{UP} \times \mathbf{n}
\]
\[
\mathbf{v} = \mathbf{n} \times \mathbf{u}
\]
Define the view volume:
(create a camera model)

```c
glMatrixMode ( GL_PROJECTION );
glLoadIdentity ( );
gluPerspective ( viewAngle, aspectRatio, N, F );
```

$N > 0, \ F > 0$
7.4 Building Viewing Matrix

View Pipeline

Canonical View Volume

- Parallel: \( x = \pm 1 \), \( y = \pm 1 \), \( z = \pm 1 \)

- Perspective: \( x = z \), \( x = -z \), \( y = z \), \( y = -z \),

  \[ z = -z_{\text{min}} , \quad z = -1 \]
**Modelview Matrix** \((M_v M_m)\):

Modeling part \((M_m)\):

- embodies all the modeling transformations for the object

Viewing part \((M_v)\):

- accounts for the WC to VC transformation set by the camera’s position and orientation

\[
M_v = \begin{bmatrix}
  u_x & u_y & u_z & d_x \\
  v_x & v_y & v_z & d_y \\
  n_x & n_y & n_z & d_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

where

\[
(d_x, d_y, d_z) = (-u \cdot \text{eye}, -v \cdot \text{eye}, -n \cdot \text{eye})
\]
Projection Matrix ($M_p$):

$$M_p = scaling2$$

* translation

* perspective transformation

* scaling1

* shearing

$$M_p = M_{s2} \ast M_t \ast M_{pt} \ast M_{s1} \ast M_{sh}$$
Shearing:

• shear so that the center of the window would coincide with \((0, 0, -N)\)

\[
M_{sh} = \begin{bmatrix}
1 & 0 & a & 0 \\
0 & 1 & b & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
a = \frac{r + l}{2N} = 0
\]

\[
b = \frac{t + b}{2N} = 0
\]
Scaling 1:

- scale so the user defined truncated view volume would coincide with the canonical view volume for perspective projection

\[ M_{s1} = \begin{bmatrix} 1/w & 0 & 0 & 0 \\ 0 & 1/h & 0 & 0 \\ 0 & 0 & 1/F & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

\[ w = F \tan(\theta/2) \cdot AR \]
\[ h = F \tan(\theta/2) \]
\[ AR = \text{aspect ratio} \]
**Perspective Transformation:**

- convert CVV for perspective projection to a quasi-CVV for parallel projection

\[
M_{pt} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \frac{F}{F-N} & \frac{N}{F-N} \\
0 & 0 & -1 & 0
\end{bmatrix}
\]
Translation:

- translate center of the quasi-CVV to the origin (0,0,0)

\[
M_t = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{2} \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
Scaling2:

- scale z-direction by 2 to get the CVV for parallel projection

\[
M_{s2} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
\[ M_p = ? \]

\[ M_p = M_{s2} \cdot M_t \cdot M_{pt} \cdot M_{s1} \cdot M_{sh} \]

\[
\begin{bmatrix}
F & 0 & 0 & 0 \\
w & F & 0 & 0 \\
0 & h & F + N & 2FN \\
0 & 0 & F - N & F - N \\
0 & 0 & -1 & 0 \\
\end{bmatrix}
\]

\[ = \frac{1}{F} \]
Clipping in Homogeneous Coordinates:

- What does $M_{pt}$ do?
Why?

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \frac{F}{F-N} & \frac{N}{F-N} \\
0 & 0 & \frac{F-N}{-1} & \frac{F-N}{0}
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= \begin{bmatrix}
x \\
y \\
\frac{Fz + N}{F - N} \\
\frac{-z(F - N)}{1}
\end{bmatrix}
\]

1. If \( z > 0 \) then \( \frac{Fz + N}{-z(F - N)} < -1 \)

2. If \( z < -1 \) then \( \frac{Fz + N}{-z(F - N)} < -1 \)

3. If \( -\frac{N}{F} < z < 0 \) then \( \frac{Fz + N}{-z(F - N)} > 0 \)

4. If \( -1 < z < -\frac{N}{F} \) then \( 0 > \frac{Fz + N}{-z(F - N)} > -1 \)
Now consider the following example:

\[ M_{pt} \] maps \( P \) and \( Q \) both into points in region \( G \)

\[
M_{pt} P = \begin{bmatrix} 0 & \frac{N/F}{F + N} \\ \frac{F - N}{F + N} & -1 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{N/F}{F - N} \\ \frac{N/F}{F - N} & 1 \end{bmatrix}
\]

\[
M_{pt} Q = \begin{bmatrix} 0 & \frac{N/F}{-2F + N} \\ \frac{F - N}{-2F + N} & 2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{N/2F}{2(F - N)} \\ \frac{-2F + N}{2(F - N)} & 1 \end{bmatrix}
\]
• If we clip the line segment \( P'Q' \) against the CVV after the \textbf{perspective division}, since \( P' \) and \( Q' \) are both to the right of the far clipping plane, we would think the entire line segment is outside the CVV and would consequently have the line discarded. But \( R'S' \) of the line is actually inside the CVV.

• The reason that this happens is because the division performed for \( P' \) changes the sign of z-component from positive to negative.

\textbf{Remedy}: perform \textit{clipping} before performing \textit{perspective division}, i.e., clip in homogeneous coordinates, then perform perspective division.
How to clip a point in homogeneous coordinates?

If we use $M_{pt} \equiv M_{s2} \ast M_t \ast M_{pt}$ as the perspective transformation

$$M_{pt} = M_{s2} \ast M_t \ast M_{pt} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & F + N & 0 \\
0 & 0 & F - N & 2N \\
0 & 0 & -1 & 0
\end{bmatrix}$$

(what is the difference between $M_{pt}$ and $M_{pt}$?)

then a point

$$(X, Y, Z, W)' = M_{pt}(x, y, z, 1)'$$

(before the perspective division) is inside the CVV for parallel projection if

$$-1 \leq \frac{X}{W} \leq 1, \quad -1 \leq \frac{Y}{W} \leq 1, \quad -1 \leq \frac{Z}{W} \leq 1$$
If \( w > 0 \), this means the boundary coordinates (BC’s) must all be positive:

<table>
<thead>
<tr>
<th>Boundary coordinate</th>
<th>homogeneous value</th>
<th>clip plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>( BC_0 )</td>
<td>( W + X &gt; 0 )</td>
<td>( x = -1 )</td>
</tr>
<tr>
<td>( BC_1 )</td>
<td>( W - X &gt; 0 )</td>
<td>( x = 1 )</td>
</tr>
<tr>
<td>( BC_2 )</td>
<td>( W + Y &gt; 0 )</td>
<td>( y = -1 )</td>
</tr>
<tr>
<td>( BC_3 )</td>
<td>( W - Y &gt; 0 )</td>
<td>( y = 1 )</td>
</tr>
<tr>
<td>( BC_4 )</td>
<td>( W + Z &gt; 0 )</td>
<td>( z = -1 )</td>
</tr>
<tr>
<td>( BC_5 )</td>
<td>( W - Z &gt; 0 )</td>
<td>( z = 1 )</td>
</tr>
</tbody>
</table>

If \( w < 0 \), then all the BC’s must be negative:

<table>
<thead>
<tr>
<th>Boundary coordinate</th>
<th>homogeneous value</th>
<th>clip plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>( BC_0 )</td>
<td>( W + X &lt; 0 )</td>
<td>( x = -1 )</td>
</tr>
<tr>
<td>( BC_1 )</td>
<td>( W - X &lt; 0 )</td>
<td>( x = 1 )</td>
</tr>
<tr>
<td>( BC_2 )</td>
<td>( W + Y &lt; 0 )</td>
<td>( y = -1 )</td>
</tr>
<tr>
<td>( BC_3 )</td>
<td>( W - Y &lt; 0 )</td>
<td>( y = 1 )</td>
</tr>
<tr>
<td>( BC_4 )</td>
<td>( W + Z &lt; 0 )</td>
<td>( z = -1 )</td>
</tr>
<tr>
<td>( BC_5 )</td>
<td>( W - Z &lt; 0 )</td>
<td>( z = 1 )</td>
</tr>
</tbody>
</table>
How to clip a line segment in homogeneous coordinates?

Use 3D Cyrus-Beck clipper:

input: \( A = (A_x, A_y, A_z, A_w) \)
\( B = (B_x, B_y, B_z, B_w) \)

\( L(t) = A + (B - A) \cdot t \)

- Compute \( BC \)'s for \( A \) and \( B \)
- Compute \emph{outcodes} for \( A \) and \( B \)
- Perform "trivial rejection" test
- Perform "trivial acceptance" test
If both tests fail, then

- for each clip plane, if \( A \) and \( B \) are on different side of the plane, find parameter of the intersection point (hit point). For instance, for \( x = 1 \) the hit point’s parameter is

\[
t = \frac{A_w - A_x}{(A_w - A_x) - (B_w - B_x)}
\]

- Then update related items’ values