Projection: why is projection necessary?
7.1 Projections

Perspective projection
7.1 Projections

Orthographic projection
Perspective 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
Orthographic projection

Three orthographic projections:

[Diagram showing three orthographic projections: front view, top view, and side view.]
Orthographic projection

Isometric projection of unit cube along direction (1, -1, -1):
7.2 Mathematics of Projections
(projections can be defined by 4x4 matrices)

Perspective projection: (not Affine, irreversible)

Projection plane is normal to the $z$ axis
7.2 Mathematics of Projections
(projections can be defined by 4x4 matrices)

\[
\begin{bmatrix}
x_p, y_p, z_p, 1 \end{bmatrix}^t = \begin{bmatrix}
x \\
\frac{-y}{z/d} \\
\frac{-z}{d} \\
-d, 1 \end{bmatrix}^t
\]

\[
= \begin{bmatrix}
x, y, z, -z/d \end{bmatrix}^t
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/d & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= M_{per}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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

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

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/d & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= M_{per}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

Orthographic (parallel) projection:
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1 \\
\end{bmatrix}
= M_{par}
\begin{bmatrix}
x \\
y \\
z \\
1 \\
\end{bmatrix}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
= M_{par}
\begin{bmatrix}
x \\
y \\
z \\
1
\end{bmatrix}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

Perspective projection: (COP is not at the origin)

\[
\begin{align*}
    x_p &= \frac{x}{1 - z/d} \\
    y_p &= \frac{y}{1 - z/d} \\
    z_p &= 0
\end{align*}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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

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

\[
= \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -1/d & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z - d \\
1
\end{bmatrix}
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & d \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -d \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\ y \\ z \\ 1
\end{bmatrix}
= M_t \, M_{per} \, M_t
\]
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

Where are the vanishing points?

• Parallel lines after perspective projection are still parallel lines if they are also parallel to the projection plane. Why?

• Parallel lines after perspective projection are no longer parallel lines if they are not parallel to the projection plane. Why?
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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
7.2 Mathematics of Projections

(projections can be defined by 4x4 matrices)

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 $AB$. 
7.3 Camera Model for Projective View

• How to create a perspective view of a scene in OpenGL?

• How to control the camera’s position and orientation in OpenGL?
7.3 Camera Model for Projective View

Conceptual model of 3D viewing:

\[ M_v M_m \rightarrow M_p \rightarrow \text{Clip} \rightarrow \text{Perspective division} \]

- Modelview matrix \( M_v M_m \)
- Projection matrix \( M_p \)
- Viewport matrix \( M_{vp} \)
- Normalized device coordinates
- Window (Device) coordinates
- Eye coordinates
- Clip coordinates
7.3 Camera Model for Projective View

Define Viewing (Eye, or Camera)
Coordinate System:
(specification of a 3D view)
(Positioning and pointing the camera)

glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
gluLookAt (eye.x, eye.y, eye.z, look.x, look.y, look.z, up.x, up.y, up.z);
7.3 Camera Model for Projective View

\[ \mathbf{n} = \text{EYE} - \text{LOOK} \]
\[ \mathbf{u} = \text{UP} \times \mathbf{n} \]
\[ \mathbf{v} = \mathbf{n} \times \mathbf{u} \]
7.3 Camera Model for Projective View

Define the view volume:
(create a camera model)

```c
glMatrixMode ( GL_PROJECTION );
glLoadIdentity ( );
gluPerspective ( viewAngle, aspectRatio, N, F );
```
7.3 Camera Model for Projective View

\[ \text{v} \]

\[ \text{n} \]

\[ \text{u} \]

\[ N > 0, F > 0 \]

near plane  

far plane  

EYE  

\[ \theta \]  

\[ N \]  

\[ F \]  

CS Dept, Univ of Kentucky
7.4 Building Viewing Matrix

View Pipeline

\[ M_v M_m \rightarrow M_p \rightarrow \text{Clip} \rightarrow \text{Perspective division} \]

- Modelview matrix
- Projection matrix
- Clip coordinates
- Perspective division
7.4 Building Viewing Matrix

Canonical View Volume

- Parallel: \( x = \pm 1, \ y = \pm 1, \ z = \pm 1 \)
- Perspective: \( x = z, \ x = -z, \ y = z, \ y = -z, \ z = -z_{\min}, \ z = -1 \)
7.4 Building Viewing Matrix

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
7.4 Building Viewing Matrix

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

where

\[
( d_x, d_y, d_z ) = ( -u \cdot \text{eye} , -v \cdot \text{eye} , -n \cdot \text{eye} )
\]
7.4 Building Viewing Matrix

**Projection Matrix**\( (M_p) \):

\[
M_p = scaling2 \\
* \text{translation} \\
* \text{perspective transformation} \\
* scaling1 \\
* shearing
\]

\[
M_p = M_{s2} \ast M_t \ast M_{pt} \ast M_{s1} \ast M_{sh}
\]
7.4 Building Viewing Matrix

Shearing: so that window center would coincide with (0, 0, -N)

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

\[
a = \frac{r + l}{2N} \quad b = \frac{t + b}{2N}
\]
Scaling1: so user defined truncated view volume would coincide with the canonical view volume for perspective projection

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

\[ w = F \tan(\theta/2) \ AR \]
\[ h = F \tan(\theta/2) \]
\[ AR = \text{aspect ratio} \]
7.4 Building Viewing Matrix

**Perspective Transformation:** convert CVV for perspective projection to a quasi-CVV for parallel projection

\[
M_{pt} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & \frac{F}{F-N} & \frac{N}{F-N} \\
0 & 0 & -1 & 0 \\
\end{bmatrix}
\]
7.4 Building Viewing Matrix

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

\[
M_t = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1/2 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
7.4 Building Viewing Matrix

Scaling2: scale z-direction by 2 to get the CVV for parallel projection

\[ \mathbf{M}_{s2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
7.4 Building Viewing Matrix

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

\[
M_p = \frac{1}{F} \begin{bmatrix}
F & 0 & 0 & 0 & 0 \\
0 & F & 0 & 0 & 0 \\
0 & 0 & F + N & 0 & 0 \\
0 & 0 & F - N & F - N & 2FN \\
0 & 0 & -1 & 0 & F - N
\end{bmatrix}
\]
7.5 GL_PROJECTION & GL_MODELVIEW

GL_PROJECTION:
- applied to every point that comes after it

GL_MODELVIEW:
- applied to every point in a particular model
GL_PROJECTION

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1, 1, -1, 1, -1.0, 1.0);
glTranslate(100, 100, 100);
glRotateF(45, 1, 0, 0);

Gl_Projection_Matrix = identity_matrix * orthographic_matrix * Translation_matrix * Rotation_matrix
GL_MODELVIEW

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1, 1, -1, 1, -1.0, 1.0);
glTranslate(camera_x, camera_y, camera_z);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslate(box_x, box_y, box_z);
// draw box here

glLoadIdentity();
glTranslate(bottle_x, bottle_y, bottle_z);
// draw bottle here
GL_MODELVIEW

For the box:

\[
\text{Box\_Vertices\_Matrix} = \text{Projection\_Ortho\_Matrix} \times \text{Projection\_Translation\_Matrix} \times \text{Box\_Translation\_Matrix}
\]

For the bottle:

\[
\text{Bottle\_Vertices\_Matrix} = \text{Projection\_Ortho\_Matrix} \times \text{Projection\_Translation\_Matrix} \times \text{Bottle\_Translation\_Matrix}
\]
7.6 Clipping in Homogeneous Coordinates

What does $M_{pt}$ do?

$A \rightarrow G$
$B \rightarrow E$
$C \rightarrow F$
$D \rightarrow G$
7.6 Clipping in Homogeneous Coordinates

Why?

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

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

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

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

If \( 1 < z < \frac{N}{F} \) then \( 0 > \frac{Fz + N}{-z(F-N)} > -1 \)
7.6 Clipping in Homogeneous Coordinates

Now consider the following example:

$M_{pt}$ maps $P$ and $Q$ both into points in region $G$
7.6 Clipping in Homogeneous Coordinates

- If line segment $P'Q'$ is clipped against the CVV after the perspective division, since $P'$ and $Q'$ are both to the right of the far clipping plane, the clipping algorithm would think the entire line segment is outside the CVV and would have the line segment discarded. But $R'S'$ 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.

**Remedy:** perform clipping before performing perspective division, i.e., clip in homogeneous coordinates, then perform perspective division.
7.6 Clipping in Homogeneous Coordinates

- Basic idea:

7.6 Clipping in Homogeneous Coordinates

Why? a point (before the \texttt{perspective division}) is inside the CVV for parallel projection if

\[-1 \leq \frac{X}{W} \leq 1, \quad -1 \leq \frac{Y}{W} \leq 1, \quad -1 \leq \frac{Z}{W} \leq 1\]
Clipping in Homogeneous Coordinates


<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>W+X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-X</td>
<td></td>
<td>x=1</td>
</tr>
<tr>
<td>W+Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-Y</td>
<td></td>
<td>y=1</td>
</tr>
<tr>
<td>W+Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-Z</td>
<td></td>
<td>z=1</td>
</tr>
</tbody>
</table>
### 7.6 Clipping in Homogeneous Coordinates

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>
7.6 Clipping in Homogeneous Coordinates

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>
Clip a Line Segment in Homogeneous Coordinates

How to clip a line segment in homogeneous coordinates?

Use Cyrus-Beck clipper:

Input:

\[ A = \left( A_x, A_y, A_z, A_w \right) \]
\[ B = \left( B_x, B_y, B_z, B_w \right) \]
\[ L(t) = A + (B - A) * t \]
Clip a Line Segment in Homogeneous Coordinates

- Compute $BC$’s for A and B
- Compute outcodes for A and B
- Perform “trivial rejection” test
- Perform “trivial acceptance” test
Clip a Line Segment in Homogeneous Coordinates

If A and B are on different sides of $x=1$, then compute parameter of the intersection point as follows:

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

Then update related items’ values
Clip a Line Segment in Homogeneous Coordinates

For \( x=1 \), we consider:

\[
(A_w - A_x) + [(B_w - B_x) - (A_w - A_x)] \ast t = 0
\]

Solving it to get:

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