Perspective Projection

- Projection – map the object from 3D space to 2D screen

Perspective()
Frustum()
Perspective Projection: Classical

Based on similar triangles:

\[
\frac{y'}{y} = \frac{N}{-z}
\]

\[
y' = y \times \frac{N}{-z}
\]
Perspective Projection: Classical

- So \((x^*, y^*)\) projection of point, \((x, y, z)\) unto near plane \(N\) is given as:

\[
(x^*, \ y^*) = \left( \frac{N}{-z}, \frac{N}{-z} \right)
\]

- Numerical example:

Q. Where on the viewplane does \(P = (1, 0.5, -1.5)\) lie for a near plane at \(N = 1\)?

\[
(x^*, \ y^*) = \left( \frac{1 \times \frac{1}{1.5}}{1}, \frac{0.5 \times \frac{1}{1.5}}{1.5} \right) = (0.666, 0.333)
\]
Pseudodepth

- Classical perspective projection projects \((x,y)\) coordinates to \((x^*, y^*)\), drops \(z\) coordinates

  - But we need \(z\) to find closest object (depth testing)!!!
Perspective Transformation

- **Perspective transformation** maps actual z distance of perspective view volume to range \([-1 \text{ to } 1]\) (Pseudodepth) for canonical view volume.

We want perspective Transformation and NOT classical projection!!

Set scaling \(z\):

\[\text{Pseudodepth} = az + b\]

Next solve for \(a\) and \(b\).
We want to transform viewing frustum volume into canonical view volume.
Perspective Transformation using Pseudodepth

\[(x^*, y^*, z^*) = \left( \frac{N}{-z}, \frac{N}{-z}, \frac{az + b}{-z} \right)\]

- Choose \( a, b \) so as \( z \) varies from \textbf{Near} to \textbf{Far}, pseudodepth varies from \(-1\) to \(1\) (canonical cube)

- Boundary conditions
  - \( z^* = -1 \) when \( z = -N \)
  - \( z^* = 1 \) when \( z = -F \)
Transformation of z: Solve for a and b

- Solving:
  \[ z^* = \frac{az + b}{-z} \]

- Use boundary conditions
  - \( z^* = -1 \) when \( z = -N \) \( \ldots \ldots \) (1)
  - \( z^* = 1 \) when \( z = -F \) \( \ldots \ldots \) (2)

- Set up simultaneous equations
  \[
  -1 = \frac{-aN + b}{N} \Rightarrow -N = -aN + b \ldots \ldots (1)
  \]
  \[
  1 = \frac{-aF + b}{F} \Rightarrow F = -aF + b \ldots \ldots (2)
  \]
Transformation of z: Solve for a and b

\(- N = -aN + b \ldots \ldots (1)\)

\(F = -aF + b \ldots \ldots (2)\)

- Multiply both sides of (1) by -1

\(N = aN - b \ldots \ldots (3)\)

- Add eqns (2) and (3)

\(F + N = aN - aF\)

\(\Rightarrow a = \frac{F + N}{N - F} = \frac{-(F + N)}{F - N} \ldots \ldots (4)\)

- Now put (4) back into (3)
Transformation of $z$: Solve for $a$ and $b$

- Put solution for $a$ back into eqn (3)

\[ N = aN - b \ldots \ldots (3) \]

\[ \Rightarrow N = \frac{-N(F + N)}{F - N} - b \]

\[ \Rightarrow b = -N - \frac{-N(F + N)}{F - N} \]

\[ \Rightarrow b = \frac{-N(F - N) - N(F + N)}{F - N} = \frac{-NF - N^2 - NF + N^2}{F - N} = \frac{-2NF}{F - N} \]

- So

\[ a = \frac{-(F + N)}{F - N} \]

\[ b = \frac{-2FN}{F - N} \]
What does this mean?

- Original point \( z \) in original view volume, transformed into \( z^* \) in canonical view volume

\[
z^* = \frac{az + b}{-z}
\]

- where

\[
a = \frac{-(F + N)}{F - N}
\]

\[
b = \frac{-2FN}{F - N}
\]
Homogenous Coordinates

- Want to express projection transform as 4x4 matrix
- Previously, homogeneous coordinates of \( P = (P_x, P_y, P_z) \) => \( (P_x, P_y, P_z, 1) \)
- Introduce arbitrary scaling factor, \( w \), so that
  \[ P = (wP_x, wP_y, wP_z, w) \]
  \text{ (Note: } w \text{ is non-zero)}
- For example, the point \( P = (2, 4, 6) \) can be expressed as
  - \( (2, 4, 6, 1) \)
  - or \( (4, 8, 12, 2) \) where \( w = 2 \)
  - or \( (6, 12, 18, 3) \) where \( w = 3 \), or….
- To convert from homogeneous back to ordinary coordinates, first divide all four terms by \( w \) and discard 4\text{th term}
Perspective Projection Matrix

- Recall Perspective Transform

\[
(x^*, y^*, z^*) = \left( \frac{N}{x-z}, \frac{N}{y-z}, \frac{az+b}{-z} \right)
\]

- We have:

\[
x^* = \frac{N}{x-z} \quad y^* = \frac{N}{y-z} \quad z^* = \frac{az+b}{-z}
\]

- In matrix form:

\[
\begin{pmatrix}
N & 0 & 0 & 0 \\
0 & N & 0 & 0 \\
0 & 0 & a & b \\
0 & 0 & -1 & 0
\end{pmatrix}
\begin{pmatrix}
w x \\
w y \\
w z \\
w
\end{pmatrix}
= 
\begin{pmatrix}
w N x \\
w N y \\
w (az+b) \\
-w z
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\frac{x N}{-z} \\
\frac{N}{y-z} \\
\frac{az+b}{-z} \\
\frac{-z}{1}
\end{pmatrix}
\]

Perspective Transform Matrix  
Original vertex  
Transformed Vertex  
Transformed Vertex after dividing by 4\textsuperscript{th} term
Perspective Projection Matrix

\[
\begin{pmatrix}
N & 0 & 0 & 0 \\
0 & N & 0 & 0 \\
0 & 0 & a & b \\
0 & 0 & -1 & 0
\end{pmatrix}
\begin{pmatrix}
wP_x \\
wP_y \\
wP_z \\
w
\end{pmatrix} =
\begin{pmatrix}
wNP_x \\
wNP_y \\
w(aP_z + b) \\
-wP_z
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
x \frac{N}{-z} \\
y \frac{N}{-z} \\
a \frac{az + b}{-z} \\
1
\end{pmatrix}
\]

\[
a = \frac{-(F + N)}{F - N} \quad b = \frac{-2FN}{F - N}
\]

- In perspective transform matrix, already solved for \(a\) and \(b\):
- So, we have transform matrix to transform \(z\) values
Perspective Projection

- Not done yet!! Can now transform z!
- Also need to transform the $x = (\text{left, right})$ and $y = (\text{bottom, top})$ ranges of viewing frustum to $[-1, 1]$
- Similar to glOrtho, we need to translate and scale previous matrix along $x$ and $y$ to get final projection transform matrix
- we translate by
  - $-(\text{right} + \text{left})/2$ in $x$
  - $-(\text{top} + \text{bottom})/2$ in $y$
- Scale by:
  - $2/(\text{right} - \text{left})$ in $x$
  - $2/(\text{top} - \text{bottom})$ in $y$
Perspective Projection

- Translate along x and y to line up center with origin of CVV
  - \(-\frac{\text{right} + \text{left}}{2}\) in x
  - \(-\frac{\text{top} + \text{bottom}}{2}\) in y
- Multiply by translation matrix:

\[
\begin{pmatrix}
1 & 0 & 0 & -\frac{\text{right} + \text{left}}{2} \\
0 & 1 & 0 & -\frac{\text{top} + \text{bottom}}{2} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
Perspective Projection

- To bring view volume size down to size of CVV, scale by
  - \( \frac{2}{\text{right} - \text{left}} \) in x
  - \( \frac{2}{\text{top} - \text{bottom}} \) in y

- Multiply by scale matrix:

\[
\begin{pmatrix}
\frac{2}{\text{right} - \text{left}} & 0 & 0 & 0 \\
0 & \frac{2}{\text{top} - \text{bottom}} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Scale size down along x and y
Perspective Projection Matrix

Scale

\[
\begin{pmatrix}
\frac{2}{\text{right} - \text{left}} & 0 & 0 & 0 \\
0 & \frac{2}{\text{top} - \text{bottom}} & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Translate

\[
\begin{pmatrix}
1 & 0 & 0 & -(\text{right} + \text{left}) / 2 \\
0 & 1 & 0 & -(\text{top} + \text{bottom}) / 2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Previous Perspective Transform Matrix

\[
\begin{pmatrix}
N & 0 & 0 & 0 \\
0 & N & 0 & 0 \\
0 & 0 & a & b \\
0 & 0 & -1 & 0
\end{pmatrix}
\]

Final Perspective Transform Matrix

\[
\begin{pmatrix}
\frac{2N}{x_{\text{max}} - x_{\text{min}}} & 0 & \frac{\text{right} + \text{left}}{\text{top} - \text{bottom}} & 0 \\
0 & \frac{2N}{\text{top} - \text{bottom}} & 0 & 0 \\
0 & 0 & -\frac{F + N}{F - N} & -2FN \\
0 & 0 & -1 & 0
\end{pmatrix}
\]

\text{glFrustum(left, right, bottom, top, N, F)} \quad N = \text{near plane, F = far plane}
After perspective transformation, viewing frustum volume is transformed into canonical view volume.
Geometric Nature of Perspective Transform

a) Lines through eye map into lines parallel to z axis after transform
b) Lines perpendicular to z axis map to lines perp to z axis after transform
Normalization Transformation

Original clipping volume

Original object

COP

New clipping volume

Distorted object projects correctly

$z = -x$

$z = x$

$z = -\text{near}$

$z = -1$

$x = -1$

$x = 1$
References