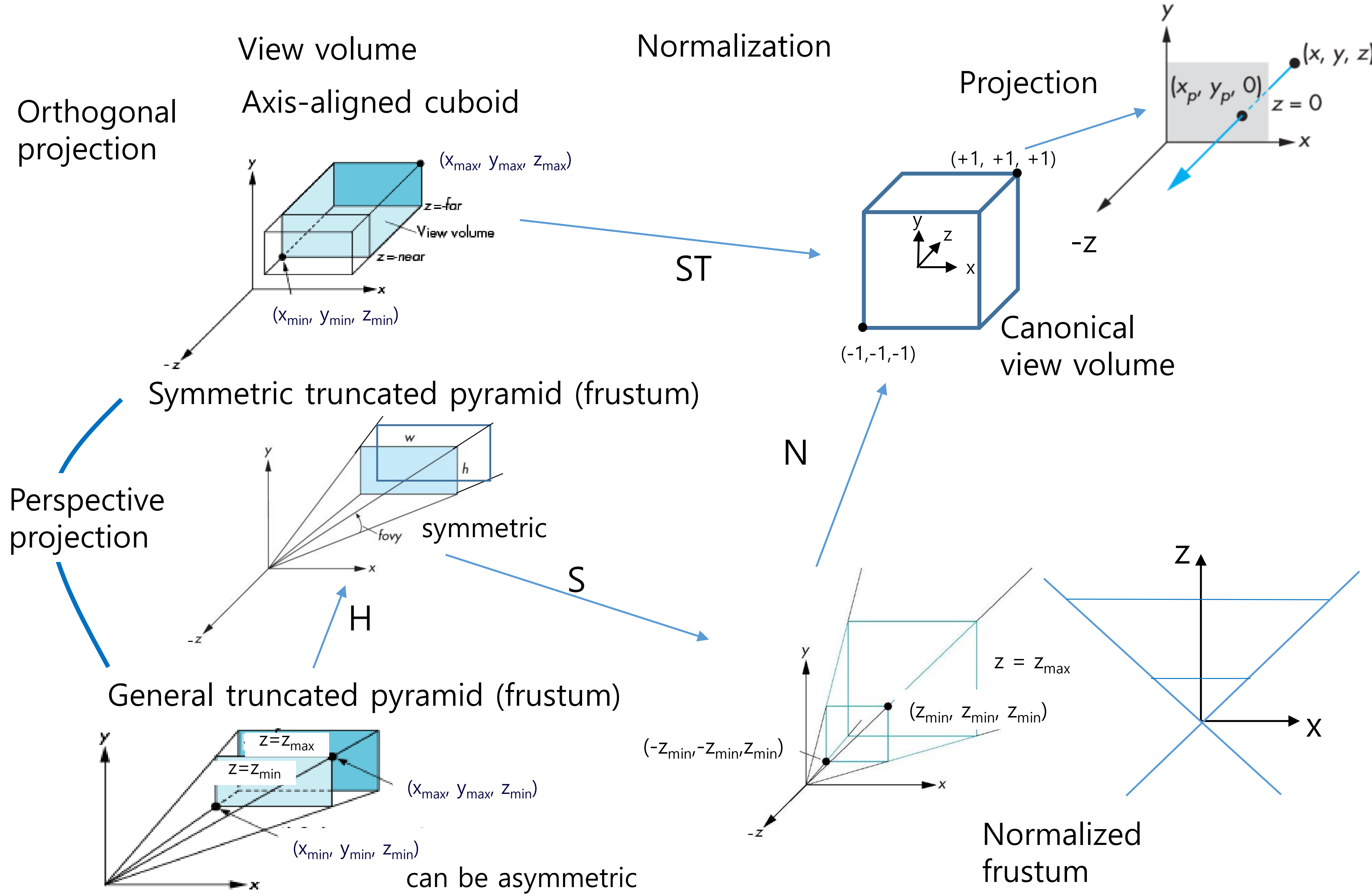
LEC20: Perspective Projection

Ku-Jin Kim
School of Computer Science & Engineering
Kyungpook National University

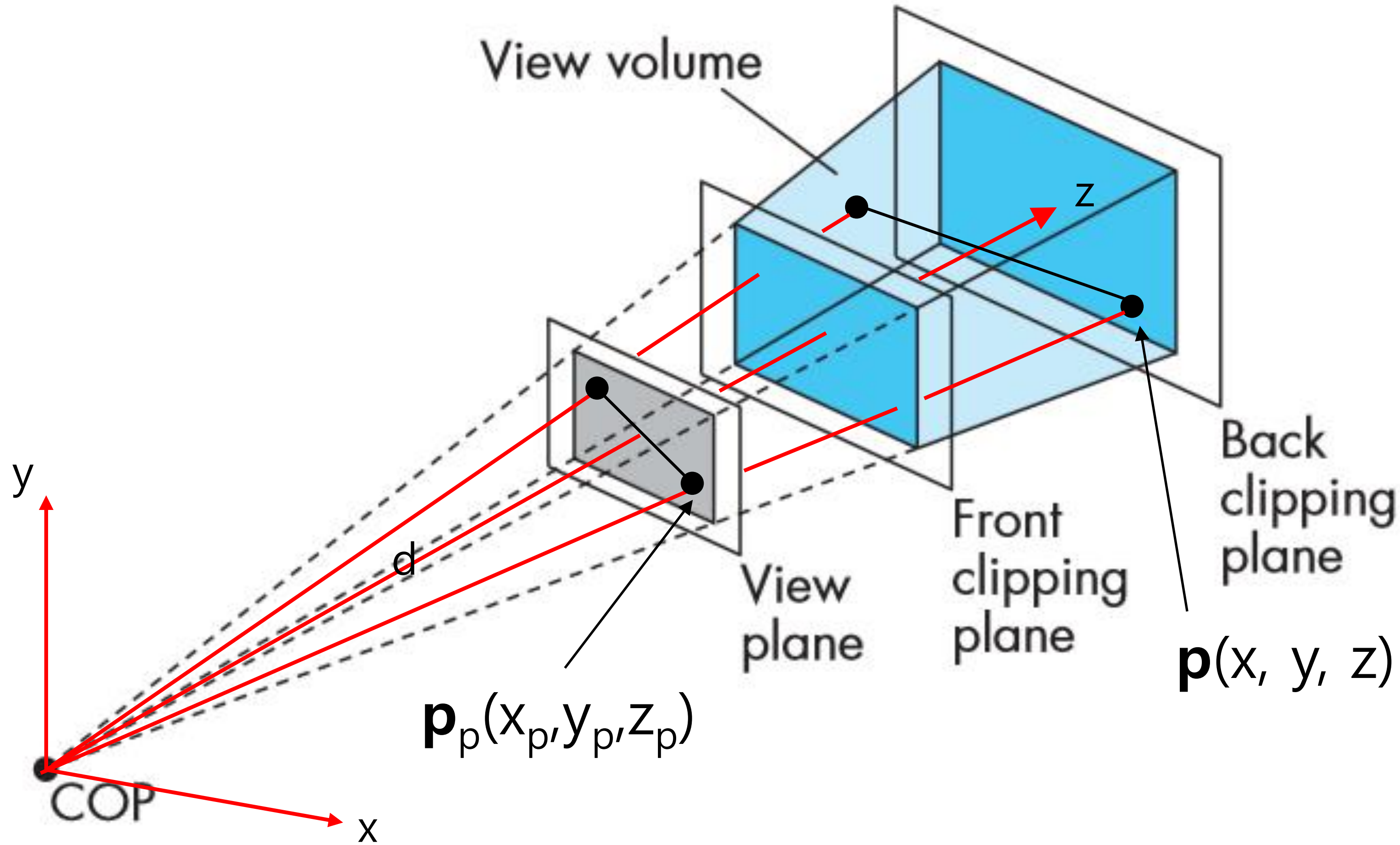
Contents

- Perspective normalization
- Perspective division
- Perspective projection matrices

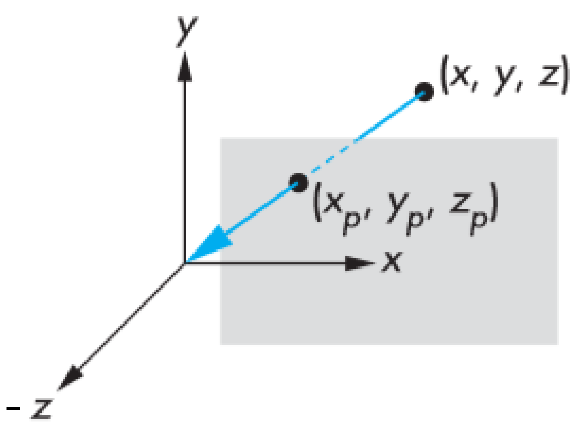
Projection Matrix Construction Overview

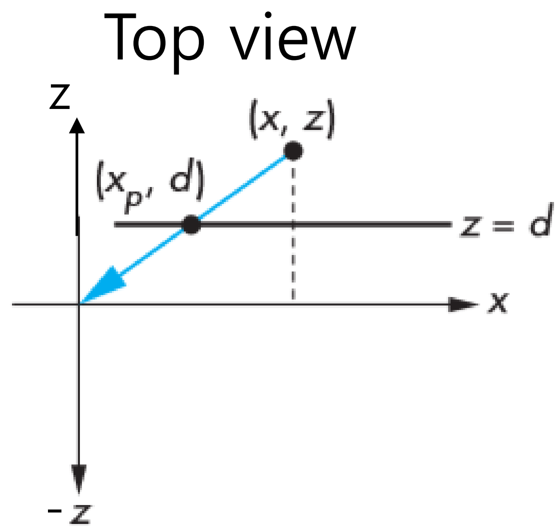


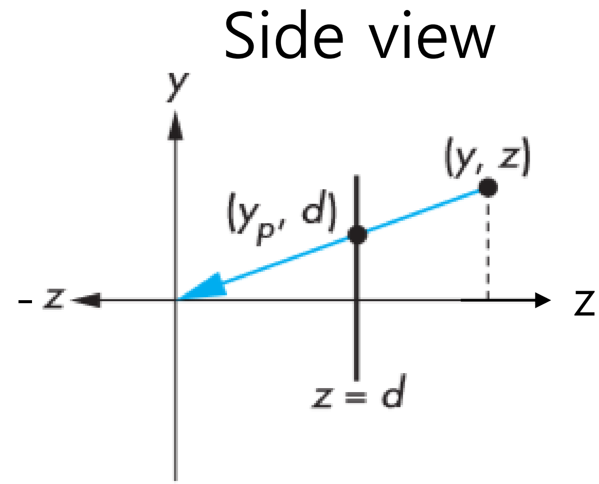
Simple Perspective Transformation (1)



Simple Perspective Transformation (2)







- d : distance from COP to view plane
  - $x : x_p = z : d$
  - $y : y_p = z : d$
- Projected point $\mathbf{p}_p$
  - $x_p = \dfrac{x}{z/d}$ $\quad y_p = \dfrac{y}{z/d}$ $\quad z_p = d$
- Not Affine transformation & irreversible

Simple Perspective Transformation (3)

- New homogeneous coordinate form
  - Rather than representing point as **p**(x, y, z, 1), we represent **p** as **q**

$$\mathbf{p} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

q = M **p**

$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Simple Perspective Transformation (4)

- Perspective division
  - divide the components of the point by w component
- When $\mathbf{q}'$ is the point derived after applying perspective division to $\mathbf{q}$, $\mathbf{q}'$ is the same as the projected point $\mathbf{p}_p$

projected point $\mathbf{p}_p$

$$\frac{1}{z/d}\mathbf{q} = \frac{1}{z/d}\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \mathbf{q}' = \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \quad \text{since} \quad x_p = \frac{x}{z/d}, \quad y_p = \frac{y}{z/d}, \quad z_p = \frac{z}{z/d} = d$$

w

Simple Perspective Transformation (5)

- If view plane at z = 1, with COP at the origin, the matrix M is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

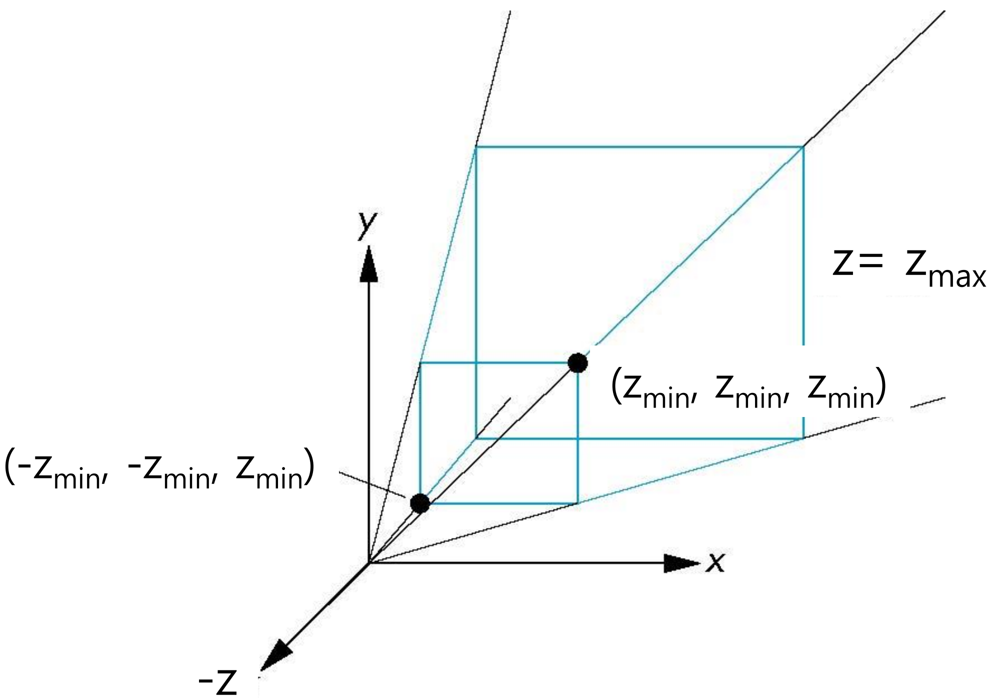
Simple Perspective Transformation (6)

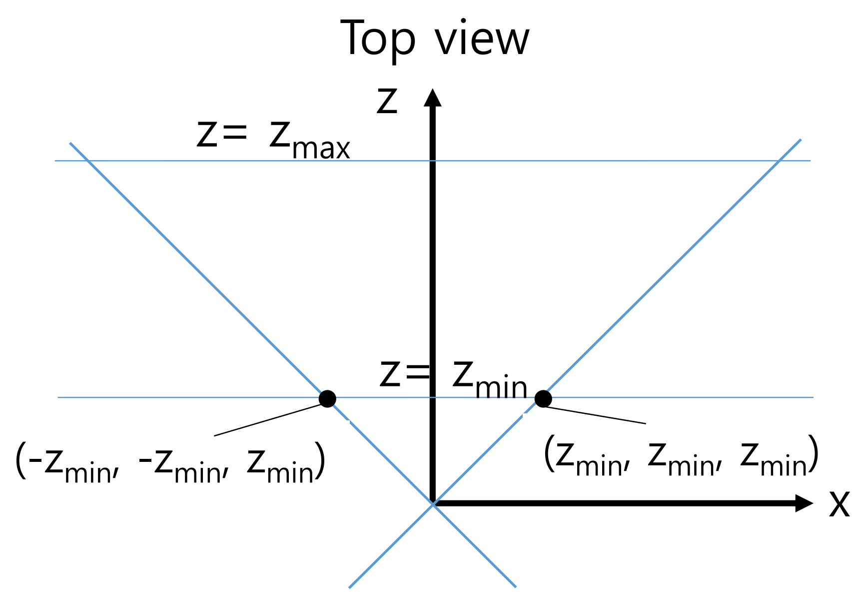
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} \xrightarrow{\text{perspective division}} \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

- Perspective division is a default process in graphic pipeline

Normalized Frustum

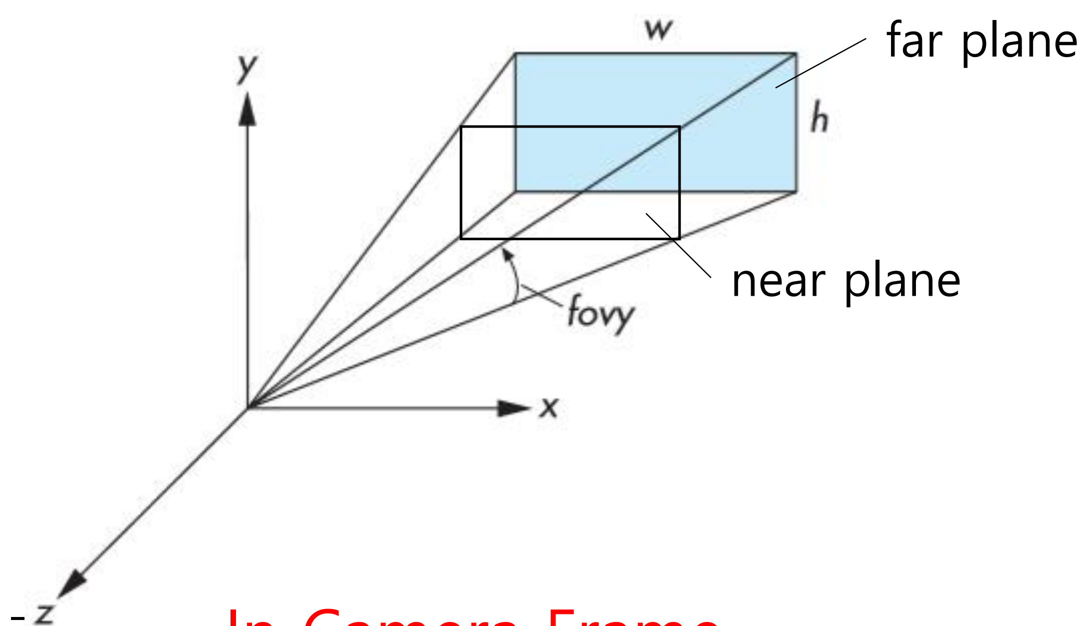
- Not official term. This term is limited to this course only.
- Consider a simple perspective with the COP at the origin, the near clipping plane at $z = z_{min}$, and a 90 degree field of view determined by the planes
  - $x = \pm z$, $y = \pm z$, $z = z_{min}$, $z = z_{max}$





OpenGL Perspective Viewing (1)

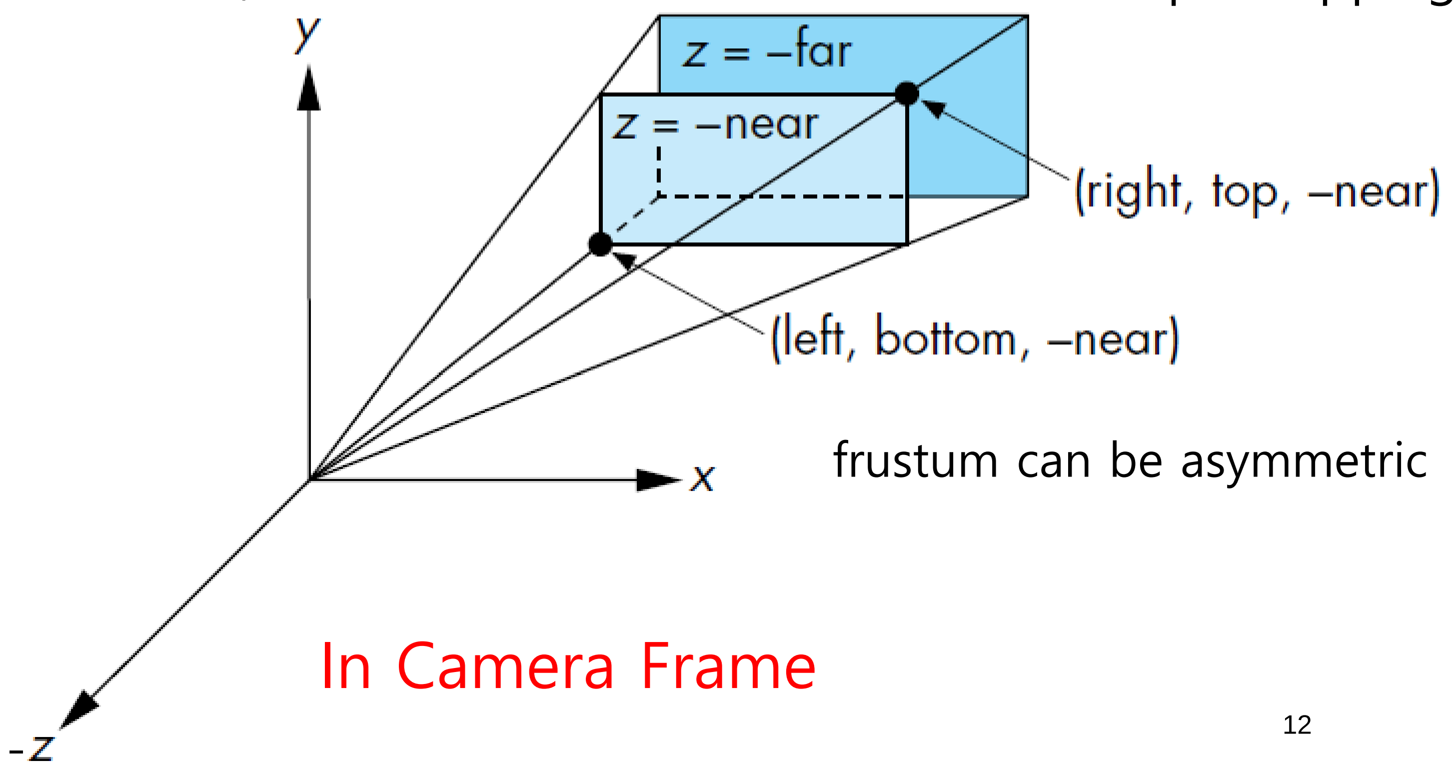
- Deprecated glu functions
  - gluPerspective (fovy, aspect, near, far)
    - fovy: field of view angle in the y direction
    - aspect: aspect ratio of width and height
    - near, far : distances to the near and far depth clipping planes



In Camera Frame

OpenGL Perspective Viewing (2)

- Deprecated gl function
  - glFrustum (left, right, bottom, top, near, far)
    - left, right : coordinate for the left and right vertical clipping planes
    - bottom, top : coordinates for the bottom and top horizontal clipping planes
    - near, far : distances to the near and far depth clipping planes

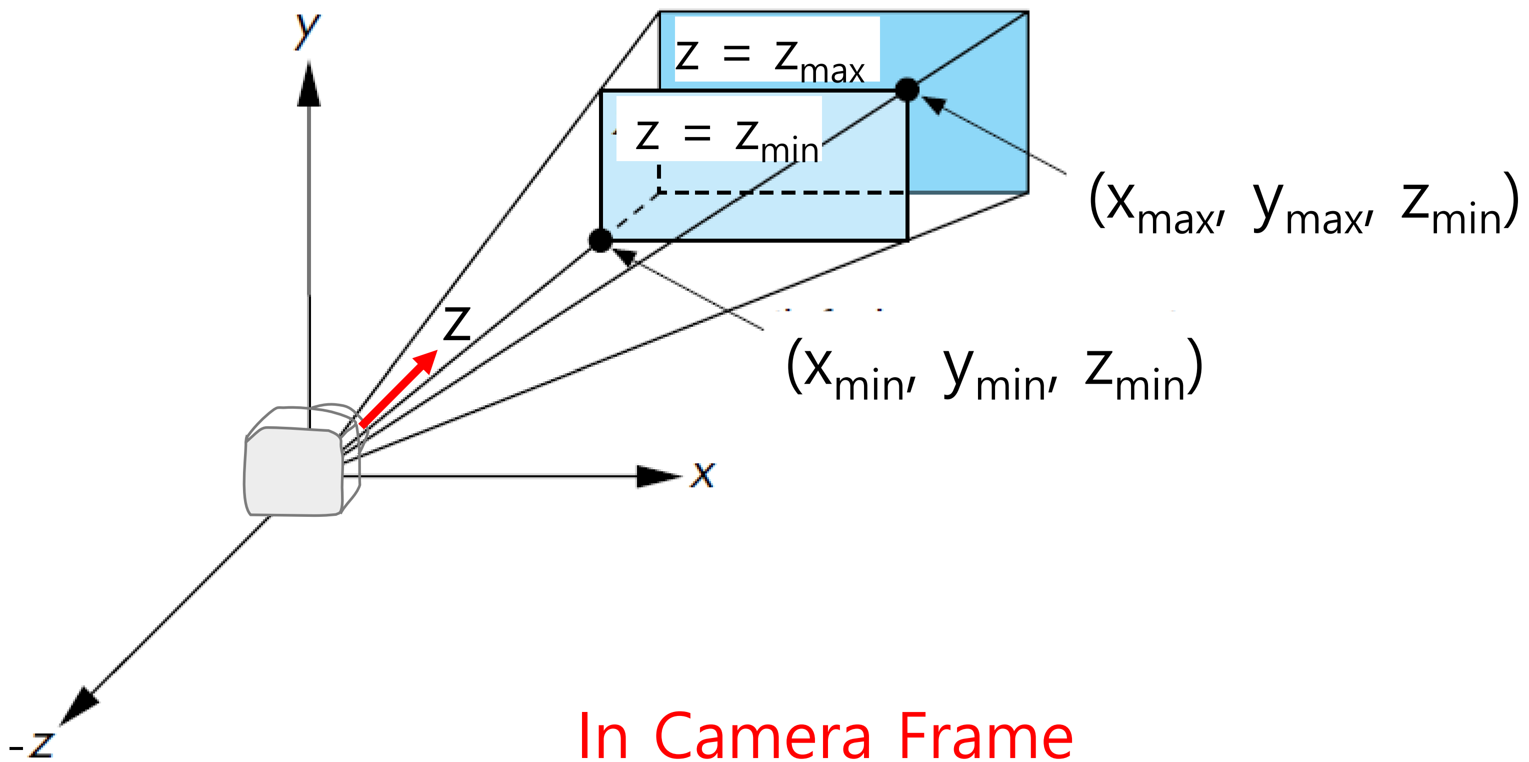


frustum can be asymmetric

In Camera Frame

Perspective Projection

- We will construct
  - myFrustum(m_{persp}, x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})
  - m_{persp}: perspective projection matrix
  - $x_{min} < x_{max}$, $y_{min} < y_{max}$, $0 < z_{min} < z_{max}$
  - Both of symmetric & asymmetric frustums



In Camera Frame

Perspective Projection Matrix for Symmetric Frustum

- Given symmetric view frustum
  - 1) a scaling to get the normalized frustum
  - 2) perspective normalization

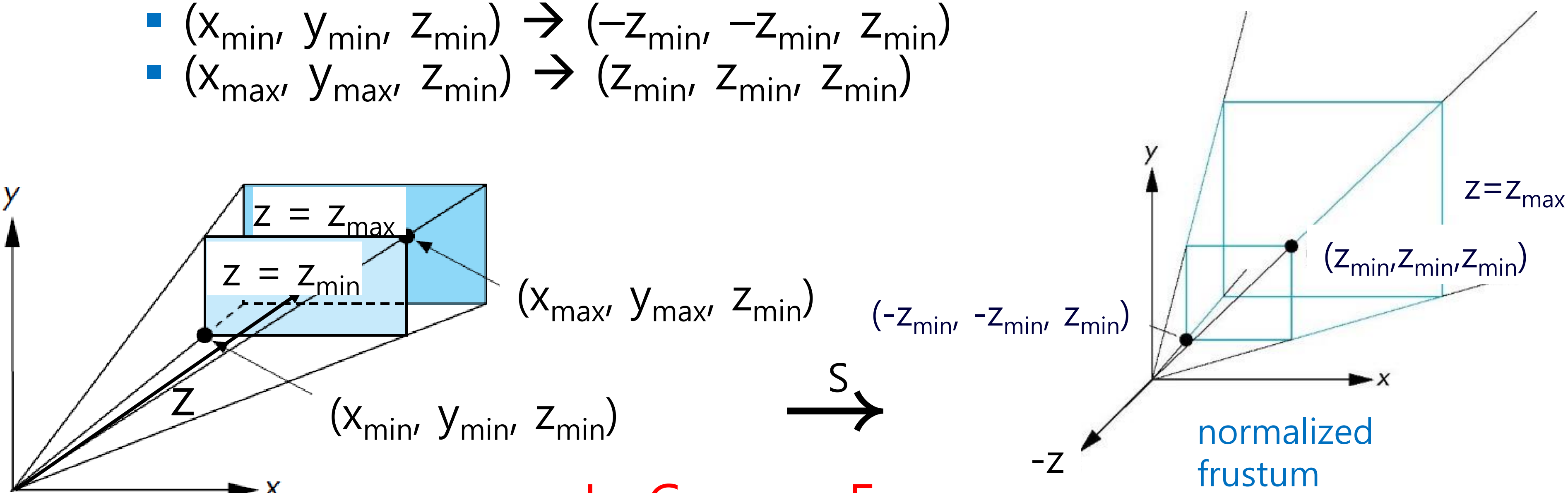
$$\mathbf{m}_{sym} = \mathbf{NS}$$

perspective normalization

scale

Perspective Projection for Symmetric Frustum (1)

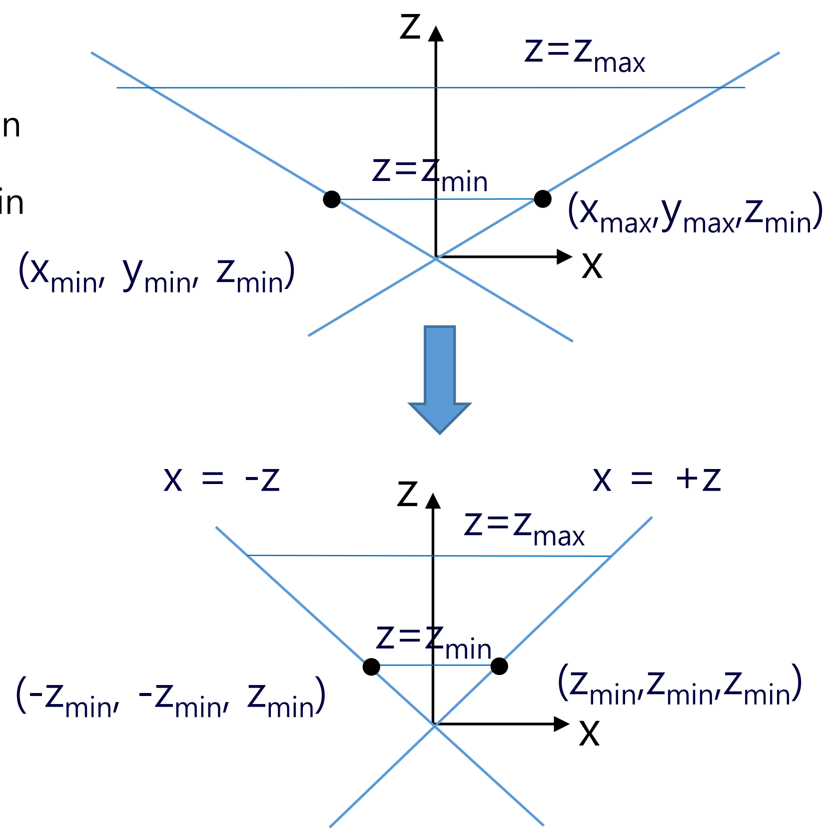
- Given symmetric view frustum (symmetric about z-axis),
  - $|x_{min}| = |x_{max}|$, $|y_{min}| = |y_{max}|$
- 1) Compute S matrix for scaling to form a normalized frustum
  - $(x_{min}, y_{min}, z_{min}) \rightarrow (-z_{min}, -z_{min}, z_{min})$
  - $(x_{max}, y_{max}, z_{min}) \rightarrow (z_{min}, z_{min}, z_{min})$



In Camera Frame

Perspective Projection for Symmetric Frustum(2)

- Scaling matrix
  - Scale the length ($x_{max} - x_{min}$) to $2z_{min}$
  - Scale the length ($y_{max} - y_{min}$) to $2z_{min}$

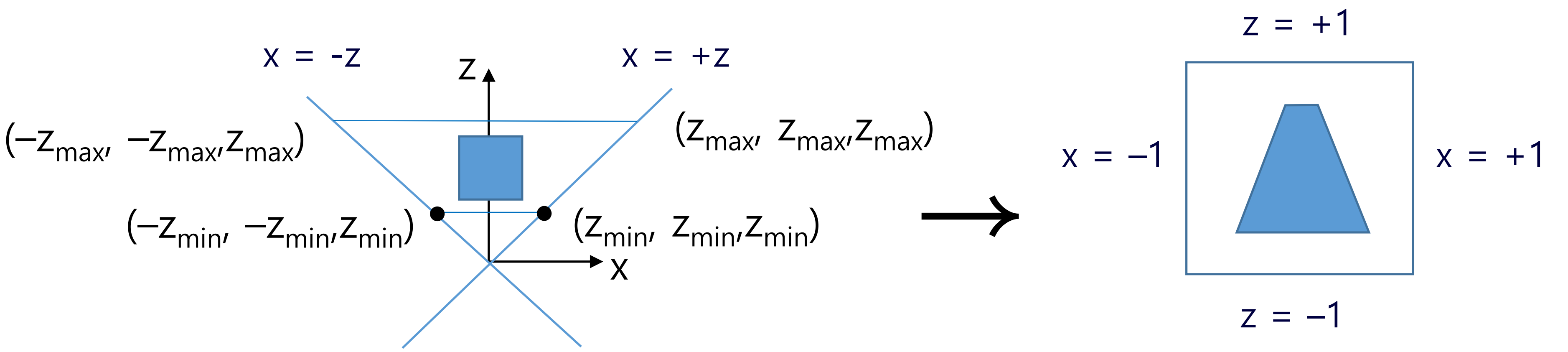


$$S = \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ex) $x_{min} = -4$, $y_{min} = -3$, $z_{min} = 2$
$x_{max} = 4$, $y_{max} = 3$, $z_{max} = 4$

Perspective Projection for Symmetric Frustum (3)

- 2) Compute perspective normalization matrix N
  - $(-z_{min}, -z_{min}, z_{min})$ -> (-1, -1, -1)
  - $(z_{max}, z_{max}, z_{max})$ -> (+1, +1, +1)



$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection for Symmetric Frustum(4)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -z_{min} \\ -z_{min} \\ z_{min} \\ 1 \end{bmatrix} = \begin{bmatrix} -z_{min} \\ -z_{min} \\ \alpha z_{min} + \beta \\ z_{min} \end{bmatrix} \rightarrow \begin{bmatrix} -1 \\ -1 \\ \alpha + \beta / z_{min} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z_{max} \\ z_{max} \\ z_{max} \\ 1 \end{bmatrix} = \begin{bmatrix} z_{max} \\ z_{max} \\ \alpha z_{max} + \beta \\ z_{max} \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ \alpha + \beta / z_{max} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\alpha = \frac{z_{max} + z_{min}}{z_{max} - z_{min}} \qquad \beta = \frac{-2 z_{max} z_{min}}{z_{max} - z_{min}}$$

Perspective Projection for Symmetric Frustum(5)

- Perspective projection matrix is

m_{sym} = N S

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{z_{max}+z_{min}}{z_{max}-z_{min}} & \frac{-2z_{max}z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & \frac{z_{max}+z_{min}}{z_{max}-z_{min}} & \frac{-2z_{max}z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection Matrix for Asymmetric Frustum

- myFrustum(m_{persp}, x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})
- Given asymmetric view frustum
  - 1) an initial shear to form a symmetric view frustum
  - 2) a scaling to get the normalized frustum
  - 3) final perspective normalization

$$\mathbf{m}_{persp} = \mathbf{N}\ \mathbf{S}\ \mathbf{H}$$

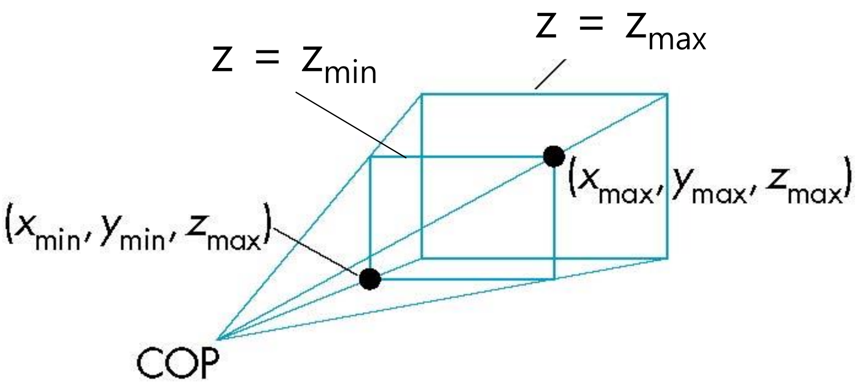
previously defined perspective normalization matrix

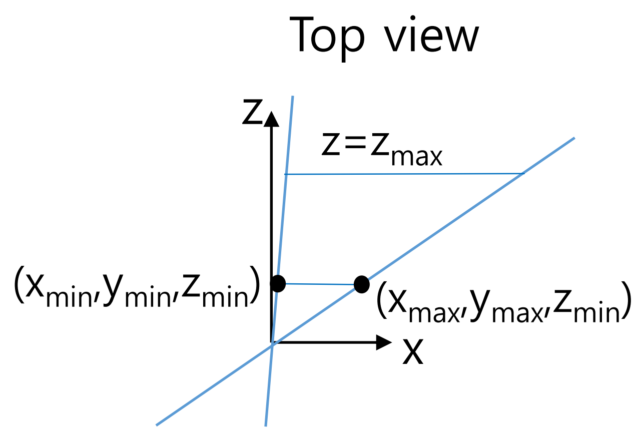
previously defined scale matrix

shear matrix

Perspective Projection for Asymmetric Frustum (1)

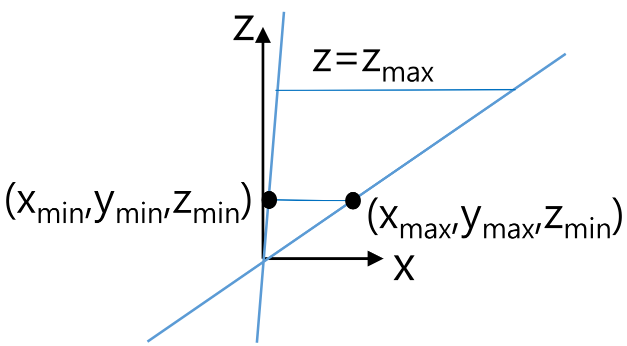
- Asymmetric frustum
  - Frustum is not symmetric about the z-axis



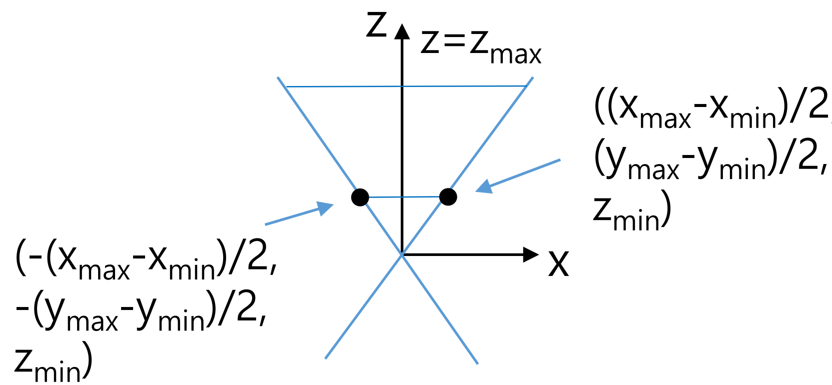


Perspective Projection for Asymmetric Frustum (2)

- 1) Shear the asymmetric frustum to a symmetric frustum
  - $(x_{min}, y_{min}, z_{min}) \rightarrow (-(x_{max}-x_{min})/2, -(y_{max}-y_{min})/2, z_{min})$
  - $(x_{max}, y_{max}, z_{min}) \rightarrow ((x_{max}-x_{min})/2, (y_{max}-y_{min})/2, z_{min})$



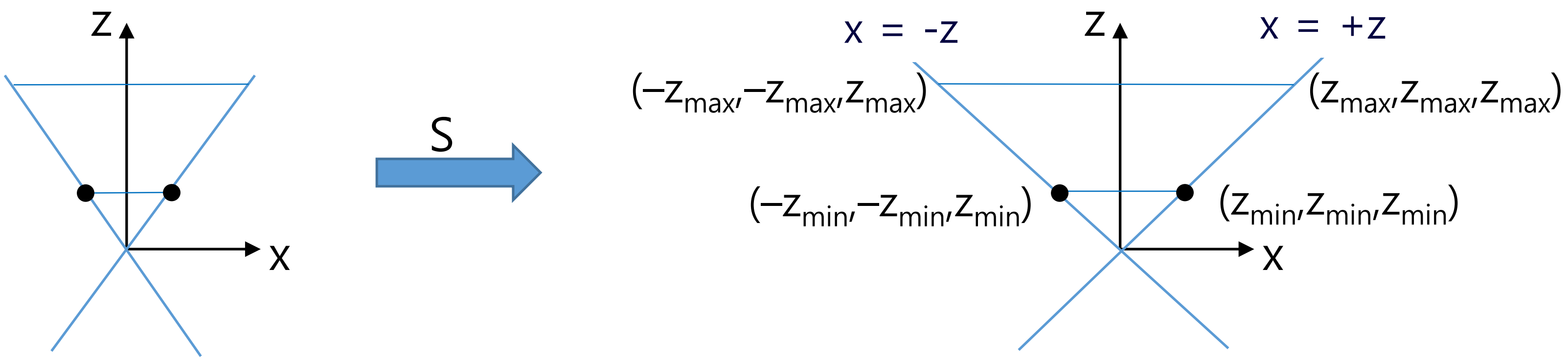
H



$$H = \begin{bmatrix} 1 & 0 & -\frac{x_{max}+x_{min}}{2z_{min}} & 0 \\ 0 & 1 & -\frac{y_{max}+y_{min}}{2z_{min}} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Perspective Projection for Asymmetric Frustum (3)

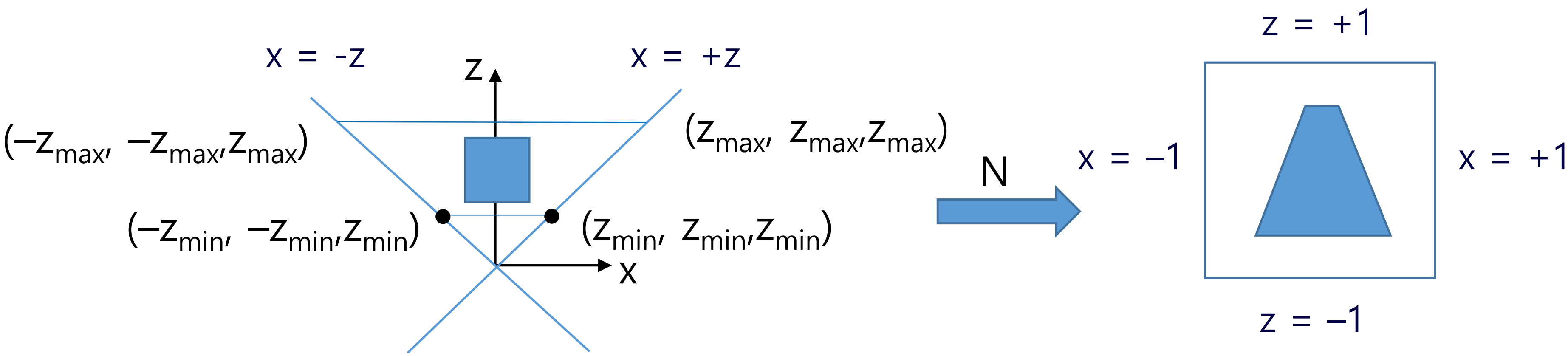
- 2) Compute S matrix for scaling to normalized frustum
  - $(x_{min}, y_{min}, z_{min}) \rightarrow (-z_{min}, -z_{min}, z_{min})$
  - $(x_{max}, y_{max}, z_{min}) \rightarrow (z_{min}, z_{min}, z_{min})$



$$S = \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Perspective Projection for Asymmetric Frustum (3)

- 3) Compute perspective normalization matrix N
  - $(-z_{min}, -z_{min}, z_{min})$ -> (-1, -1, -1)
  - $(z_{max}, z_{max}, z_{max})$ -> (+1, +1, +1)



$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{z_{max}+z_{min}}{z_{max}-z_{min}} & \frac{-2z_{max}z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection for Asymmetric Frustum (4)

- Perspective projection matrix is

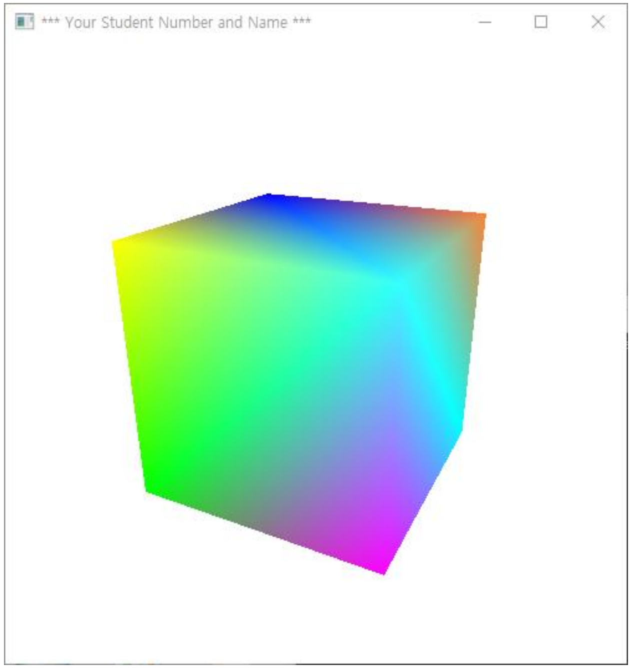
$m_{persp} = NSH$

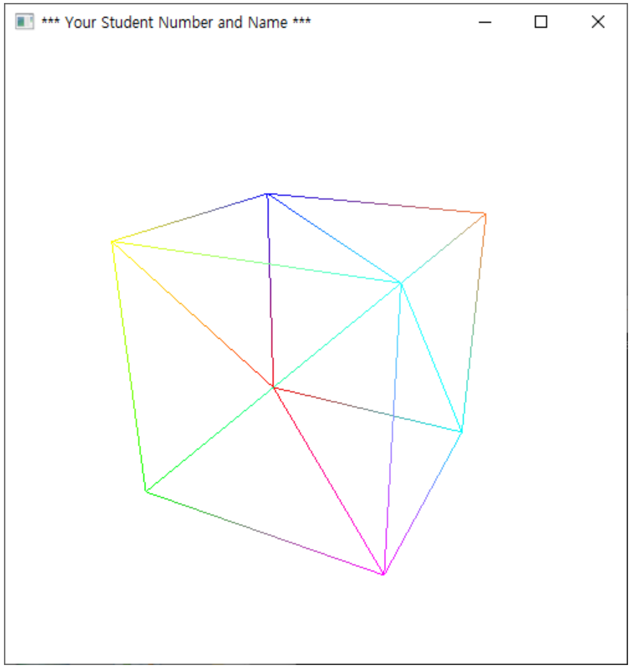
$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{z_{max}+z_{min}}{z_{max}-z_{min}} & \frac{-2z_{max}z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{x_{max}+x_{min}}{2z_{min}} & 0 \\ 0 & 1 & -\frac{y_{max}+y_{min}}{2z_{min}} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2z_{min}}{x_{max}-x_{min}} & 0 & -\frac{x_{max}+x_{min}}{x_{max}-x_{min}} & 0 \\ 0 & \frac{2z_{min}}{y_{max}-y_{min}} & -\frac{y_{max}+y_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & \frac{z_{max}+z_{min}}{z_{max}-z_{min}} & \frac{-2z_{max}z_{min}}{z_{max}-z_{min}} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Projection Example (1)

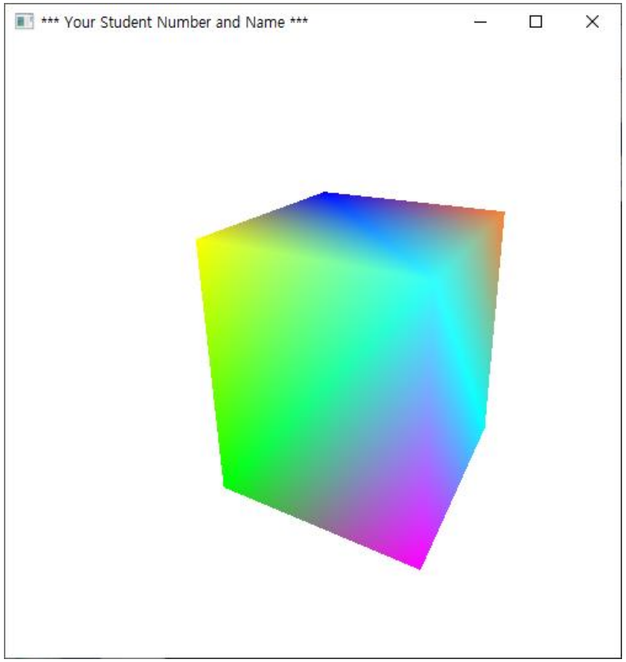
- Axis-aligned cube
  - Center: (0, 0, 0)
  - Length of all edges: 1
- LookAt
  - eye[4] = { 1.0, 1.0, 2.0, 1.0 },
  - at[4] = { 0.0, 0.0, 0.0, 1.0 },
  - up[4] = { 0.0, 1.0, 0.0, 0.0 };
- myFrustum
  - float xmin = -0.5, xmax = +0.5, ymin = -0.5, ymax = +0.5, zmin = 1.0, zmax = +4.0;

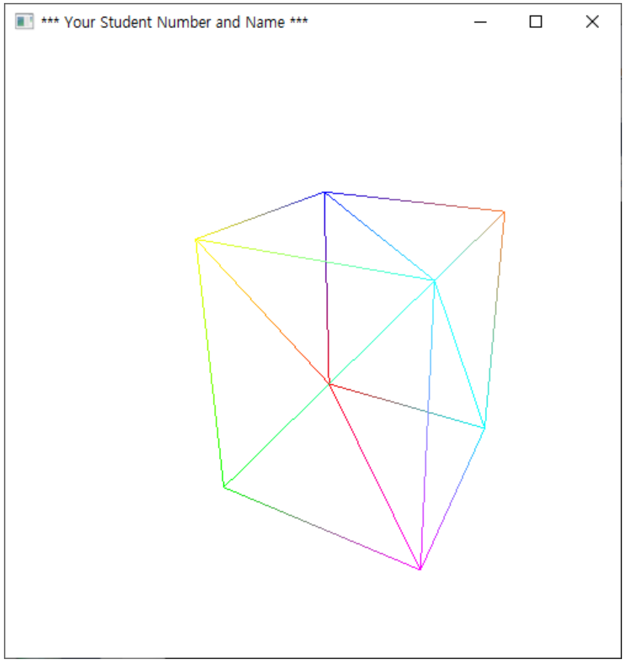




Perspective Projection Example (2)

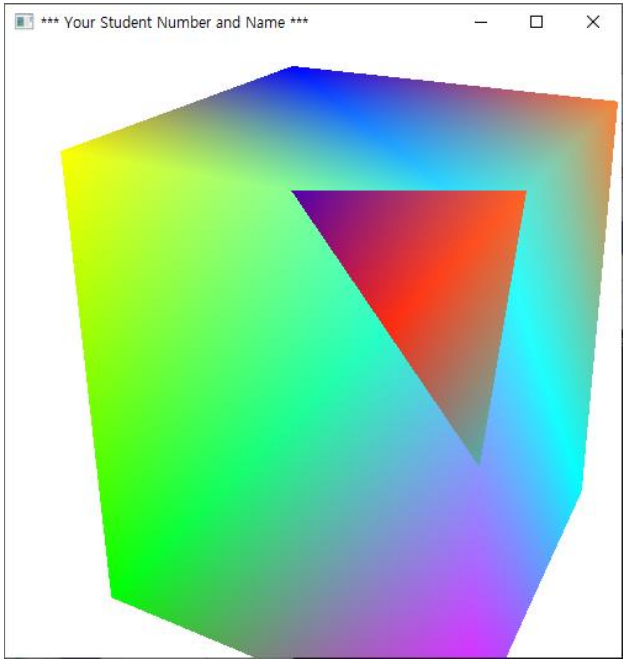
- myFrustum
  - float xmin = -0.7, xmax = +0.5, ymin = -0.5, ymax = +0.5, zmin = 1.0, zmax = +4.0;

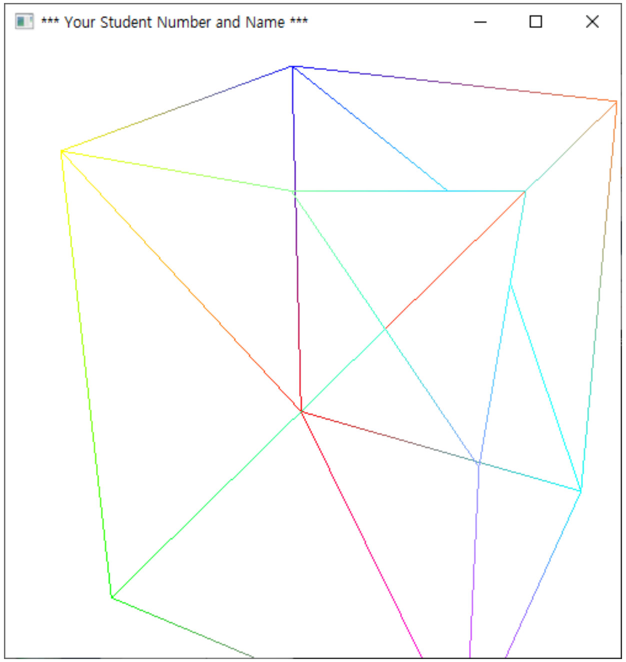




Perspective Projection Example (3)

- myFrustum
  - float xmin = -0.7, xmax = +0.5, ymin = -0.5, ymax = +0.5, zmin = 1.8, zmax = +4.0;





HW#20 Perspective Projection

- Due date: This Friday 6:00pm
- Implement myFrustum function, and call this function to generate the right image when you run the program.
- Object information is the same with HW#19 (can be found in LEC19 Lecture note board)
- Model matrix is an identity matrix
- View matrix is derived by LookAt function with parameters:
  - eye[4] = { 1.0, 1.0, 2.0, 1.0 },
  - at[4] = { 0.0, 0.0, 0.0, 1.0 },
  - up[4] = { 0.0, 1.0, 0.0, 0.0 };
- The parameters of myFrustum function:
  - float xmin = -0.7, xmax = +0.5, ymin = -0.5, ymax = +0.5, zmin = 1.8, zmax = +4.0;
- Submit .c file through LMS Homework board.